ZERO-DAY GOVERNANCE:

An (Inexpensive) Solution to the Cyber-security Problem

SANDRO GAYCKEN

Institute of Computer Science, Freie Universität Berlin

FELIX FX LINDNER

Recurity Labs GmbH, Berlin

MARCH, 2012

CYBERDIALOGUE2012

WHAT IS STEWARDSHIP IN CYBERSPACE?

Canada Centre for Global Security Studies

MUNK SCHOOL OF GLOBAL AFFAIRS

UNIVERSITY OF TORONTO

This paper investigates whether sophisticated cyber-attackers could be permanently deterred by an international governance effort directed at a continuous mass discovery of zero days. Based on strategic considerations of the risks, costs, and benefits of the zero-day development cycle and of offensive cyber-operations, we argue that such an effort could in fact cause deterrence by a lack of confidence and by denial as sophisticated cyber-operations — being highly dependent on zero days — would become significantly more expensive, risky, and uncertain. The underlying strategic conditions also demonstrate a number of economic advantages for the governance side as this effort would only have to concern itself with the rather easy and cheap process of zero day discovery instead of going for the full exploit design process, letting an effort in zero day governance appear both possible in the short-term and significantly more cost-effective than other approaches to confront sophisticated attackers. Following the core of the argument, a number of potential additional advantages can be highlighted. Vulnerability removal could outpace vulnerability creation, leading to a security saturation. Technological monocultures could be turned from a disadvantage into an advantage. The approach would be friendly to the current insecure IT-environment, thus more acceptable for states, for the IT-industry, and IT-using industries. Privacy would not be affected. There will be a small, albeit clear contribution to arms control. And finally, due to the obvious global security benefits, the low costs, and no impact on Internet regime questions, it should be possible to get a joint international effort to pull this through, at least among some willing nations. As potential problems, we will address the institutionalization, the design of the disclosure process, the creation of the necessary workforce, and the associated problem of proportional patching.[1]

THE REAL SHAPE OF THE CYBER-SECURITY PROBLEM

WRONG RISKS AND REAL RISKS

The view of the cyber-security problem is still lopsided. The public. the press, many policy-makers, and some researchers continue to worry more about harmless incidents than about real threats. The DoS-attacks on Estonia in 2007 or the recent Stratfor incident are but two examples. Both cases were widely reported as highly critical and relevant, yet no one was able to point out why exactly. The reason for this lack of explanation was neither the technical complexity of the topic, nor any high demand for secrecy. It was simply the lack of any real damage to be reported. The attacks on Estonia were very common activists' attacks on a few, rather irrelevant websites with negligible

[1] Please note that this paper is intended as an inspirational piece. Some of the involved antecedents can not be objectively confirmed at present, but have to be treated as open research questions instead. We highlighted these antecedents accordingly. Most of these uncomfirmed relations were derived from general impressions of the community of practitioners. The authors want to thank John Mallery from MIT, Chris Demchak from the Naval War College, Andrew Cushman from Microsoft, Andreas Könen and Michael Hange from the BSI, and the Harvard Cyber Policy seminar for useful objections and comments. The views expressed reflect the views of the authors and do not necessarily reflect those of the Research Group Secure Identity at the Freie Universität Berlin.

damage caused by the disruption of services.[2] The loss of customer data from Stratfor resulted in some financial damage as activists diverted money by using stolen credit card credentials, but this was pretty common and soon managed by the credit card companies.[3] A number of reasons can be given for the unnaturally prolonged life of this misperception. Low-level attacks are more visible than the rarely disclosed high-level ones, making a better showcase for the press, for politicians, and the industry.[4] The inherent uncertainties about what can be done by hacking seem to justify dystopian speculations, despite the fact that unsophisticated attackers usually lack the necessary resources and expertise to cause critical damage in high-security environments. And unsophisticated attackers can be managed—so products can be sold, political measures can be initiated, and results can be presented to buyers and voters. But unavailable websites and teenagers disclosing customer data are no crises—this is hype. Not a single reported instance ever caused damage even remotely close to a large natural catastrophe, a terrorist attack, or to an unconventional attack by nuclear, chemical, or biological weapons — categories frequently drawn when cyber risks are compared.[5]

Sophisticated attackers on the other hand are quite different.[6] In their current frequency, they are a recent phenomenon. And they are much more dangerous. They can hire and combine a diverse range of cutting-edge experts. They can use spies and insiders to gather intelligence, to do reconnaissance, or to deploy attacks. They can invest significantly more money in attack design, including procuring test labs or even hardware companies. These advantages enable them to go after any kind of target, to attack without having to use the Internet, and to be invisible, unidentifiable, and unattributable.[7] Sophisticated attackers are also better at exploiting their attacks. They can get a diverse range of very attractive returns from their targets, including devastating damages at these targets.[8] And that they exist is an open secret in the security world. High-value targets such as the stock exchange or militaries already suffer on a regular basis from these attackers. Stuxnet and

[2] In Estonia, a small number of government and banking websites went offline for a few hours each day over a period of a few weeks. While annoying and causing minor financial damages, taking down a government website is not equal to taking down a government. It's more like taking down a very large poster of a government. This should not entice people to talk about war-like conditions. However, the press, Estonia, NATO, and a few professionals nonetheless did so, such as an EU-report which claimed that "a case can be made that the disruptions in Estonia 2007 could possibly have been equivalent to a 'kinetic attack'" (in A. Klimburg et al.: "Cybersecurity and Cyberpower: Concepts, Conditions and Capabilities for Action within the EU," p. 52, http://www.oiip.ac.at/home/home-detail/article//cybersecurity-and-cyberpower-concepts-conditions-and-capabilities-for-action-within-the-eu.html).

[3] Stratfor, a think tank with a self-proclaimed proficiency in all things cyber, has been hacked by teenage activists from Anonymous, who in turn published Stratfor's (easily accessible and non-encrypted) customer data. The stolen data float around on the Internet (e.g., here: http://pastebin.com/f7jYf5Wd). Yet again, while annoying, losing customer addresses and credit card information hardly amounts to a public crisis. Nonetheless, media worldwide portrayed the story in the most dramatic terms. It was the number-one news item in Germany that day.

[4] California already implemented stronger regulation in 2002 (California Civil Code 1798.82), and the Securities and Exchange Commision (SEC) issued new guidelines on 13 October 2011 regarding stronger requirements to report cyber-attacks. More such regulations are to follow, in the US and in Europe, which will discuss stronger disclosure and liabilty regulation in the LIBE commission by the end of March 2012.

[5] This context was highlighted for instance in the UK national security strategy of 2010:http://www.direct.gov.uk/prod_consum_dg/groups/dg_digitalassets/@dg/@en/documents/digitalasset/dg_191639.pdf.

[6] For a detailed analysis, see Sandro Gaycken, *Cyberwar – Das Internet als Kriegsschauplatz* (Munich: Open Source Press, 2010).

[7] See Sandro Gaycken, "Die sieben Plagen des Cyberwar," In R Schmidt-Radefeldt & C Meissler, C. (eds.), *Automatisierung und Digitalisierung des Krieges* (Berlin: Forum Innere Führung, 2012).

[8] See Sandro Gaycken, *Cyberwar — Vom digitalen Angriff zum realen Ausnahmezustand* (Munich: Random House, 2012).

some larger espionage incidents were some of the first public cases, but they are really just the visible tips of an iceberg.[9] And apart from these already ongoing incidents, almost every country or larger criminal organization wants to have more and more sophisticated hackers. The world should worry about this threat rather than about the unperilous but prominent low-level variant.[10]

HIGH CYBERSECURITY IS NOT A TECHNICAL PROBLEM, IT'S AN ECONOMICAL PROBLEM

Worrying about sophisticated attackers is no mean feat. Due to the attribution problem, the main thrust has to be generated by passive technical security, and the high-security IT environment needed for this is demanding. Coherent, paradigmatically guided research on this kind of technology has not fully started and much of what is currently suggested is not fully explored. However, it seems to be common ground that such an IT environment would have to be a different kind of IT altogether. The very complex, very large, very networked, ever growing, ever faster version seems fundamentally indefensible against sophisticated attackers, and most approaches to high security suggest a renunciation of these paradigms in one way or another. Trusted platforms obtained by separation kernels, microkernels, transparent stacks, or formal verification, trusted paths generated by information flow control, enhanced cryptography, or trusted assurance through more transparency, logging, monitoring, and control at all layers — all these approaches aim or require to disentangle complexity, and to reduce size, speed, and networking.[11] Less is more in high security.

The IT industry and parts of the computer science community might object and point to the long track record of incremental innovations solving the longer and larger track of insecurities on a case-to-case basis while maintaining the dominant paradigms. But these incremental approaches never won the rat race—in fact, they seem to have made it a rat race in the first place, to the constant advantage of the attacker. To implement sufficiently sophisticated security, it seems quite reasonable that complexity and omnifunctionality would have to be limited, networking and centralization would have to be reduced, and that some design paradigms of IT security — such as perimeter security and reactive, "ex post facto" security — might have to be reconsidered too.

But these are strong demands, and many deem them impossible. The reasons are economical. A reform towards high-security IT environments seems to be unaffordable. First of all, these systems still need to be developed. They exist in theory, not as products. That will take some time and incentives. Second, all insecure systems including much of their periphery and some staff would have to be replaced by a slimmed-down highly secure system with a slimmed-down highly secure periphery and a new workforce capable of managing these new systems.

[9] Stuxnet was a super-worm. It significantly and undetectably sabotaged the Iranian nuclear program.

[10] See Sandro Gaycken, "Get Cyber Real! " *Survival*, 53, no. 5 (2011).

[11] Some research even suggests turning away from basic paradigms of computing languages. See Len Sassaman, Meredith Patterson, Sergey Bratus, Michael Locasto, and Anna Shubina, *Security Applications of Formal Language Theory* (Dartmouth Computer Science Technical Report TR2011-709, 2011, http://www.cs.dartmouth.edu/~sergey/langsec/papers/langsec-tr.pdf); and Len Sassaman, Meredith Patterson, Sergey Bratus, and Anna Shubina, "The Halting Problems of Network Stack Insecurity," *Login* 26, no. 6 (2011): http://static.usenix.org/publications/login/2011-12/openpdfs/Sassaman.pdf.

Third, the highly secure IT would be more costly and less efficient in its operation. Every machine would need to delegate some amount of computational power to control itself; many processes would need to be checked before processing; and the loss of omnifunctionality, of networking and of centralization would require some compensation. In other words, less can be done, what can be done would take longer, and a lot of tasks would have to be reassigned to humans who would have to reorganize their institutions to accommodate these new modes of work. That might be good news for employment, but it's bad news for costs. Finally, some systems might even suffer safety or functionality problems that are too significant to be compensated at all. The best example is the stock exchange, which at present could not possibly compensate for the loss of number-crunching power and communication speed following from a secure IT-environment.

So this solution is not much of a solution. Apart from the fact that the holy grail of progress seems to be besmirched by a regress to less technology, it seems too expensive. Many of the cost-benefit-considerations are necessarily still rather theoretical, but vendors already fear they might never be able to sell such high-security products, turning to sufficiently profitable, common IT security products. Even if the high-security environment is different on a number of counts compared to the regular IT-security market and its well-known economical problems[12] — high-security buyers often have to bear more risk, have to behave maximally responsible, are by nature less picky about security, and tend towards expensive products rather than cheap ones in the light of uncertainty[13] — an honest reform would likely be far too costly. Thus an interesting question arises: Can there be a way out without changing the predominant paradigms?

0-DAY GOVERNANCE

THE RESOURCE OF SOPHISTICATED ATTACKERS

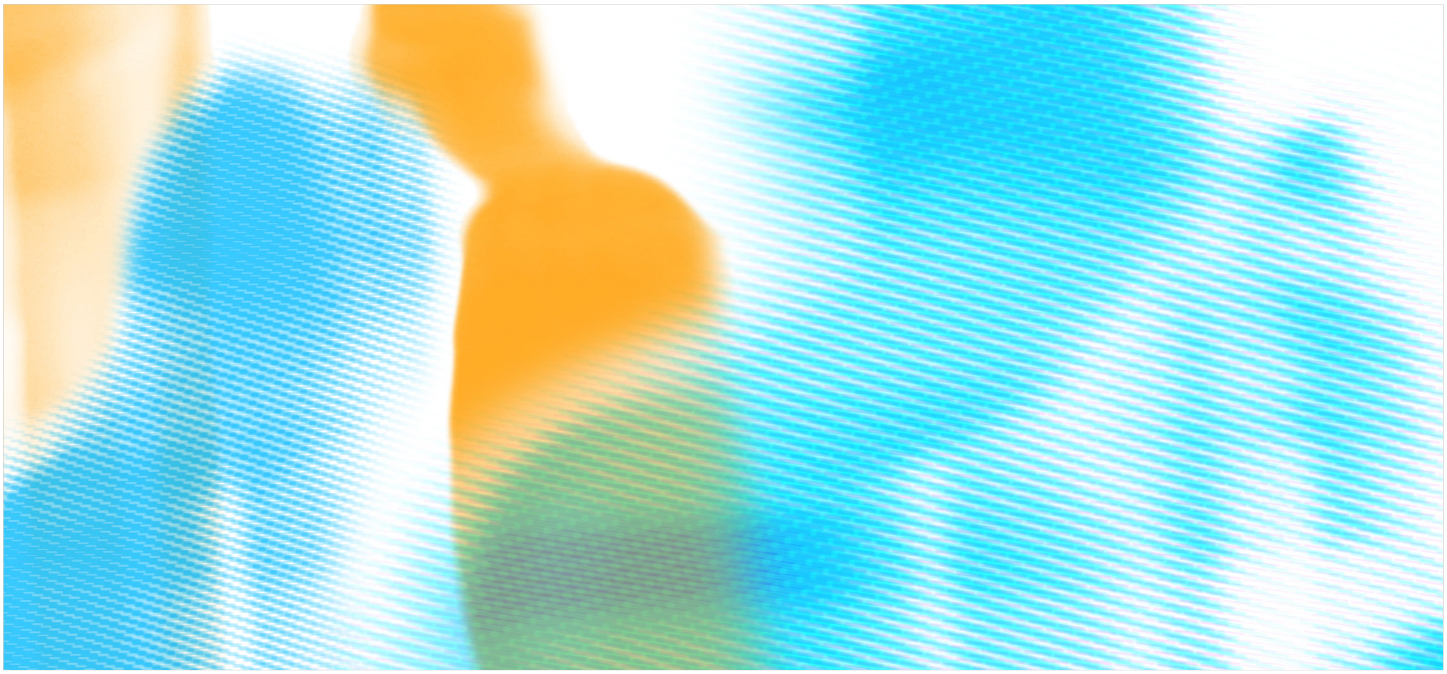
One unappreciated fact about sophisticated attackers is that they thrive on a resource. This resource is the so-called "0-day" ("0" as in zero, but spoken as "Oh"-day). There is no agreed-upon definition of 0-days by now, but speaking roughly, an 0-day is the exploitation (or, in hacker parlance, an "exploit") of a security flaw unknown to the community (thus the term — known for zero days, as the community usually counts the days by which a flaw is known).[14]

Prima facie, a number of vectors can be used to attack a system. Such vectors are needed as a point of entry and frequently also to operate

[12] See for instance, Tyler Moore and Ross Anderson, "Internet Security," in Martin Peitz, and Joel Waldfogel (eds.), *The Oxford Handbook of the Digital Economy* (Oxford University Press, 2012 forthcoming).

[13] The market behaves like the already researched "lemon market" (see Ross Anderson, "Why Information Security is Hard—An Economic Perspective," in *Proceedings of the 17th Annual Computer Security Applications Conference* [Washington DC: IEEE Computer Society, 2001], 358-65), but on a different scale. High-security buyers tend to buy the cheapest variant out of a field of specified expensive alternatives which have to comply to some explicit and certain implicit standards, in turn usually incentivizing vendors to supply the minimal set of expected specifications with a little extra, a situation which could probably be described as a "golden lemon market."

[14] This preliminary definition could be varied in respect to how novel and innovative it is, where the exploit is situated in the system (in the network, the operating system, or an application for an instance), or in the attack (to penetrate, to escalate privileges, to disable security functions, etc.), or along the lines defined by technical standardization authorities such as NIST or ISO. However, for our purposes, this kind of precision is not needed. We use the term simply to describe a somehow unknown security problem.



inside the targets, to penetrate further barriers or to outsmart safety and security mechanisms. The attacker can use the human user, and known or unknown technological vulnerabilities. In common IT environments, the human user and known technological vulnerabilities are efficient vectors, but not so in high-security environments. High-value targets care about their security in the best-possible sense. They will have implemented a good or high standard of common IT security and site safety, a high awareness in their workforce, and they will have professional teams with good practices to monitor these systems (and the workforce). This has an impact on the choice of attack vectors and on operative considerations in this environment. Enticing the human user to click something unorthodox is much less efficient because these systems usually have a number of practices and technologies in place to avoid even sophisticated attempts of social engineering. Exploiting known vulnerabilities is less likely to succeed too. This is a very common procedure in ordinary cybercrime. But a high-value target with good basic security is unlikely to be vulnerable to a known security problem. These institutions have solid patching practices and capable personnel. Of course, by now a lot of stories about high-value targets with sloppy security are known, Stratfor being a case, but also RSA, an IT-security company that was breached with simple methods in 2011. But by and large, security is usually good in these places and getting better. This leaves one vector for the sophisticated attacker to attack a high- value target. Sophisticated attackers have to use unknown technical vulnerabilities. They have to use 0-days.[15]

[15] Using the supply chain as an attack vector also has to be considered very effective at present, owing to the technical monocultures around the globe. But it is demanding. Insiders have to be hired and instructed carefully, posing a risk in themselves. And as the supply chain is a point of entry among secret services for some time already, some systems have a high level of awareness to this end already. This is expected to rise further in the future, already incentivizing vendors to pay more attention to insider detection and trusted assurance and high-value targets to aim for trusted vendors. This may mitigate the effects of supply-chain attacks over time. Many attacks will still require the use of technical vulnerabilities.

THE ECONOMICS OF 0-DAY DEVELOPMENT[16]

0-days are not readily available. It is tough work to find and develop them. It requires a lot of system-specific knowledge, experience, and hands-on fidgeting with the targeted system. The toughest part of the exploit development is to render a discovered vulnerability into a working exploit that can be implemented at the target without causing it to crash. This is a known fact about software in general. If it is sloppily designed, the system crashes. The same holds for exploits. A sophisticated attacker, however, does not want his target to crash. Any crash on such systems with professional teams will attract their attention. The defenders will suspect an attack, look closely at what happened during the crash, and might very well find the attacker. In that case, all the attacker's effort is gone, his attack halted.[17] Thus, sophisticated attackers have a maximum interest in stealth, which is realized by an exploit design process with great care. It requires the following phases:

1. In-depth intelligence and reconnaissance, mapping the target down to the very last detail, and obtaining the target's code. This phase frequently involves non-digital, physical intrusion and infiltration, including the recruitment of potential insiders;
2. Weak links, options, targets, and task have to be formulated in light of the target's structure;
3. 0-day discovery in the code obtained in all areas deemed critical for the attack;
4. Attack design in iterative phases of attack development and attack testing under near-real conditions. This entails the packaging of the discovered 0-days into a working exploit (in other words, the programming of the attack from its core to its periphery) and the design of obfuscation, including a number of false tracks, if of interest for the attacker;
5. Preparatory strikes, more intelligence, and reconnaissance or side attacks on sensors, on safety mechanisms, and other detection mechanisms of the defender might be necessary. Most security mechanisms are just as sloppily developed as everything else, thus offering enough exploitable vulnerabilities.

All of this is quite time-consuming, and because this is the working time of highly skilled experts, it is expensive time. While discovery is not that time-consuming, the testing of the exploit usually is. Most sophisticated attacks also require more than just a single 0-day. In this case, the complexity of the attack rises linearly if the attack modules do not interact, but exponentially if they do interact, since they have to be fine-tuned to each other.

For the defensive purpose discussed here, this difference between discovery and development is an important advantage for the defender. The defender needs to know only detection and some kind of mitigation in order to deal with

[16] By discussing 0-day economics, we refer to the risk economics of the discovery and design process with a particular focus on the mindset of sophisticated attackers. This is different from other approaches to vulnerability economics. These usually assess strategic incentives and the ensuing dynamics of the vulnerability market or ideal vulnerability markets (see for instance Karthik Kannan and Rahul Telang, "Economic Analysis of Market for Software Vulnerabilities," in *Third Workshop on the Economics of Information Security* (WEIS, May 2004) or *Vulnerability Life Cycles as Product Life Cycles* (2004)). See also B Schneier, "Full Disclosure and the Window of Opportunity," *Crypto-Gram*, 15 September 2000.

[17] Some cases are known in which attackers even maintained the systems they had targeted so that these systems would not crash, luring the defenders into the impression that their system was perfectly safe and didn't require any deeper inspection.

the vulnerability, while the attacker has to invest much more time, knowledge, and money to develop "weapons grade" exploits.

To give a more concrete example, a widely used exploitation for high-profile attacks can be taken into closer consideration. Out of the 639 different types of weaknesses listed by the CWE initiative of MITRE, let's consider the category "CWE-633: Weaknesses that Affect Memory," generally just called memory corruption vulnerabilities.[18]

In order to find an issue within CWE-633, the attacker can inspect the program code of the target application, either in source code (if available) or in binary form, and identify potentially insecure operations manually. Alternatively and more often, the attacker will use an automated process that feeds more or less structured random input into the application and observes its run-time behaviour. This automated process, known as "fuzzing," is used by software companies and attackers alike. The level of automation today has reached the point where the attacker will be handed individual input sets that have been found to crash the target application and have already been automatically classified as potentially exploitable by some algorithm.[19] Turning this raw finding into an exploit may be successful or not, but this is normally determined within less than twenty-four person hours of work. The resulting attack is commonly called a "proof-of-concept" exploit, meaning that it can show the working attack on a specific instance of the targeted software. The discovery process is finished at this point, and it only took a few hours or little more than a day. The discovery of more sophisticated vulnerabilities can take much longer. But on average, discovery per exploit might be well under twenty person days.

The development to follow the discovery is a rather different story. In order to turn this exploit into an attack of the quality needed for high-profile attacks, it must be tested on the complete variety of possible targets and customized according to the environment. This includes various versions of the operating system, various language configurations of the operating system, and the software itself and different versions of the target software, scenarios with and without additional security configurations, and so on. It should be obvious that this testing effort is an exponential function over the tested scenarios, and that it is a rather individual process, hard to automate. Every single test step might cause a development effort as required by the first exploit. Additionally, the exploit method used must often be revised or completely reconsidered in the light of testing results. According to Dave Aitel, the development time for a complete "weapons grade" exploit is 500 person-*days* of work.[20] In comparison with the time frame needed for discovery, this is much more.

After the design phase, the attacker is still not done. For most attacks, the operation itself is another time-consuming task. The reason is — again — the need for stealth. Attacks might have to propagate inside the target system in a quiet fashion to arrive at a critical point. Privileges might have to be escalated first to migrate from a lower level of trust to a higher one. Information usually has to be gathered and filtered carefully and patiently. Many acts of sabotage need a lot of time and patience with many small and unnoticeable or seemingly harmless minor

[18] "Common Weakness Enumeration" dictionary entry, http://cwe.mitre.org/data/slices/2000.html, accessed 21 February 2012.

[19] http://msecdbg.codeplex.com/, accessed 21 February 2012.

[20] http://www.usenix.org/events/sec11/stream/aitel/index.html, accessed 21 February 2012.

incidents slowly ascending into a catastrophe. And finally, the exfiltration or the deletion of the attack is important too, so the attacker can re-use it.[21] In any of these operational phases, anything rash might be detected. Sensors might alarm the defender, systems could crash; and the attack could be discovered and mitigated before the desired effect has been achieved.

The actual operational time will depend a lot on the individual attack. But for most cyber-attacks at this level, a certain period of activity has to be anticipated, frequently going into months. In other words, undertaking cyber operations is anything but quick. Quite unlike its portrayal in movies, high-end hacking is tedious and expensive at almost every step (which does not make good movies).

Apart from the different times needed for discovery, design, and operation, other economic factors of these different phases also have to be acknowledged. Losses have to be calculated if an 0-day is discovered prematurely or if the attack is discovered during the operation. In the first case, the loss is simply the time and other expenses that have been spent on discovery and design thus far. In the second case, the losses are more complex and higher. If an attack is discovered during an ongoing operation, it falls into the hands of the defender. This has a lot of consequences as the defender gains a lot of the attacker's know-how.

She will know:

- that she is under attack by an interested attacker, capable of hacking;
- what the attack consisted of technically, likely including all 0-days of the attacker and their packaging;
- which intelligence has been gathered about her;
- what her system's vulnerabilities are;
- which methods the attacker used and which kind of expertise he has at his disposal;
- what similar attacks might look like;
- how and where else to defend against similar attacks in the future;
- how the attack was conducted, including possible insiders;
- with a lot of luck she might have a slight chance of actually identifying the attacker because he will deem himself undetected and might not cover up his data flows yet;
- how she herself could use this attack for an attack on someone else, including targets of the alleged attacker;
- and she might disseminate this knowledge to others.

Thus, if an attack is discovered during its operation and falls into the hands of the defender, a lot of money and effort is gone, a second attack will be much harder, the whole attack team might have to be reassembled, many tactics and methods might be useless for future attacks in the same or in a different context, the whole process of intelligence gathering and reconnaissance will have to be repeated (and under much different conditions), the attack might backfire, and so on. These are significant risks and costs to be considered in planning an attack. At present, they are negligible. The chances of an 0-day being discovered during an attack are very slim. But this can change.

[21] It is a commonplace that cyber-attacks are single-use only, but this is not true. It holds only if the attack is discovered. If it can be deleted or exfiltrated, it can be re-used. Many espionage attacks have been used over the years in a large variety of targets because they have not been discovered.

0-DAY GOVERNANCE

To sum up, the more important cyber-security problem is the affordable management of sophisticated attackers, and these attackers thrive on a resource under the auspices of certain economic conditions and perceived risks.

These economics now offer an unrecognized chance to jeopardize the business models of sophisticated attackers. The basic idea is simple. If 0-days are expensive and time-consuming to develop, and if attacks become detectable and defendable and disadvantage the attacker if they're discovered prematurely, why not invest heavily in 0-day discovery? If a significant amount of 0-days can be discovered at one time, 0-day based attacks are at greater risk and more expensive.

This would have a significant impact in a very particular place. It would impact the potential attacker's *strategic mindset* — the planning process and the planner's confidence as well as the perception that the attack itself is attractive. In the mind of the attacker, cyber-attacks would be a lot less reliable on a number of counts. This change in the mindset might not discourage every attacker under every circumstance. But it would certainly discourage a large number of possible actors less prone to risk or less well resourced, and attacks on targets of only medium or uncertain value. It would function as deterrence by a lack of confidence. How well and how far this would work still remains to be assessed. But militaries, for instance, normally want high degrees of confidence, in particular if costs rise too. The rising costs and risks will also cause deterrence by denial to actors with certain defined or factual thresholds for their offensive cyber-activities. This would be progress compared to the current state of affairs. Right now, any actor at a sophisticated level — in other words, any military or any larger organization — can do anything without fear of high costs or risk.

However, with only a small number of researchers and specialized companies assessing exploitable vulnerabilities, the world is still quite far away from a significant 0-day discovery rate. Estimates suggest that a mere 250 (2011) to 500 (2009) 0-days are discovered worldwide per month.[22] In other words: there will always be enough vulnerabilities left.[23] But this is the point at which governance steps in. What if a number of capable nations join in on an effort to discover as many exploitable vulnerabilities as possible? This would be an effort worth trying — in particular because it might not be all that expensive for actors of this size. There are still some non-trivial problems, but first, a very basic idea of the math of 0-day governance can provide an impression of the numbers involved (for the sake of clarity, more detailed math will be postponed to a later paper). The current presentation is meant simply to provide a first idea.

WHICH FACTORS HAVE TO BE ACKNOWLEDGED AS INFLUENTIAL?

- Because insecurities are rather individual, any calculation has to be made per IT product. This will be a basic condition.
- Three basic variables are the amount of code (in an integral way since code is continuously expanded), the amount of exploitable flaws, and the criticality of the exploits. There are no average numbers on this. No

[22] See National Vulnerability database, NIST, http://nvd.nist.gov, accessed 23 Februrary 2012.

[23] See Eric Rescorla, "Is Finding Security Holes a Good Idea?" In IEEE *Security and Privacy*, 3, no. 1 (2005): http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1392694.

independent, methodological measurements exist. Software is very individual in this respect, depending on its development process, and different exploitable flaws are exploitable in dynamically varying degrees of criticality. But on an individual level, for systems in a certain, fixed state, sufficiently educated guesses could be undertaken.

- Some first time-oriented variables are — on the attacker's side — the time needed for the discovery of an exploit, the testing of an exploit, and the time needed for an operation. On the defender's side, critical time factors are, again, the time for the discovery of an exploit, and the time needed for its disclosure and patching.
- Damages as part of the risk equation in the attacker's planning process will play a role too.

Now two initial probabilities can be given per targeted product. The first probability p(t-d) is the likelihood that the 0-day developed by the attacker is discovered by the defender during the design phase. The second probability p(t-op) is the likelihood of the exploit being discovered during the actual operating phase. This presupposes that the operation is in fact detectable upon knowing the 0-day used to get into or abuse the system. This might not always be the case. But some exploits will be operative (in other words, needed and used during the attack), and some vulnerabilites might allow an ex post facto detection of the attack. In these cases, the risks can be calculated as given.

$$p(t\,(d)) = \frac{\text{Exploits Discovered by the Defender During Design Phase}}{\text{Exploits Possible for the Product}}$$

$$p(t\,(op)) = \frac{\text{Exploits Discovered by the Defender During Operating Phase}}{\text{Exploits Possible for the Product}}$$

Operational time may vary, depending on the attacker's intentions, but it is mostly more extensive: twice or three times the time needed for design is rather common.

Also, a joint probability can be given as:

$$p\,(t(d)+t(op)) = \frac{\text{Exploits Discovered by the Defender During Design/Operating Phase}}{\text{Exploits Possible for the Product}}$$

These numbers give an attacker some first ideas about how risky it is for him to design 0-days. In an ideal world, the exploits discovered by the defender during the design and operating phase would be almost equal to the exploits possible for the product. The attacker would not have a chance of designing an attack at all.

The reason why the first two probabilities are distinguished is the strong variance in risk associated with these two attack phases (which we mentioned earlier). As illustrated, having an attack discovered during the operating phase poses a higher risk because the damage is expected to be much higher. This can be portrayed by disjunct risk assessment, making use of the disjunct probabilities. The first risk r(d) would be the risk of an 0-day being discovered during the design phase.

$$r\,(d\,) = p(\,t\,(d\,))\times(\text{Damage by Loss of 0-Day})$$

The second risk r(op) would be the risk of an 0-day being discovered during its operating phase.

$$r\,(op) = p(t\,(op))\times(\text{Damage by Discovery of Operation})$$

With

$$(\text{Damage by Loss of 0–Days}) \subset (\text{Damage by Discovery of Operation})$$

and

$$(\text{Damage by Loss of 0–Days}) \ll (\text{Damage by Discovery of Operation})$$

Other factors and some integrals to accommodate details can be neglected for now. In an actual attack, these risks will be rather individual, varying from case to case. A lot depends on the actual capabilities of the attacker: his know-how and the necessities and sloppiness of the defender. But since many of these factors are not known in advance by the attacker, he will have to assess operational costs to justify his attack on the basis of a somewhat more agnostic risk calculation with fixed assessments for damages. The overall risk can be given as:

$$r(all) = r(d) + r(op)$$[24]

This is risk per attacked software or hardware, and pre-operational and operational risk might have to be kept separate if the values at risk do not sum up neatly. If different products are targeted at the same time, the risks might have to be calculated as a single product if the different attacks depend on each other and can be calculated as a sum if the attacks are independent of each other. Many risks will be formulated as costs necessary for design and operation, including fallback strategies and redundancies, and will mostly be quantified as (highly expensive) person-hours. A few extra costs will evolve from the necessity to buy test equipment, but this will most likely be a small sum. Other risks will evolve from the sheer loss of offensive capabilities to others once the know-how possessed exclusively is made public, or from possible escalations as nations are more likely to react more drastically to serious cyber-intrusions than in the past — even if attribution is not certain. If this risk of an escalation is unacceptable for the attacker, he will have to invest significantly more into the design of one or of multiple false flags. This is easily possible within cyber-operations. But in order to render a false flag believable for foreign security professionals, a lot of effort has to be undertaken, including the believable design of a different context. This can easily multiply operative costs a few times over.

Now the final idea to be formulated is the “exploit discovery rate” necessary to thwart attackers’ cost-benefit-calculi. First, an average attacker’s willingness to accept a certain amount of overall risk has to be formulated. It will consist of different types of risks, associated with different attack phases and different attack vectors, and it can be quantified as some number displaying a sum of money or — if escalation is to be considered — other assets:

$$r\ (acc)=\sum r_x$$

This sum could also be displayed as a continuum, encompassing a variety of risk behaviours. This way, other versions of overall risk can be formulated as functions of r(acc) in order to look at either the mean risk accepted by potential attackers or the highest risk accepted, if the goal is to discourage as many attackers as possible and not just the bulk. Assessments of the different cyber-actors’ tendencies to accept certain risks under certain circumstances remain to be made — another open research question. But many sophisticated attackers in average situations might already be deterred by a 70 percent probability of discovery during the design phase and a 30 percent probability of discovery during operation.

[24] Some risks could depend on each other, but a discussion of this scenario shall be postponed for now.

The acceptable risk will allow an easy, preliminary formulation of the exploit discovery rate. The bottom line is simple: so many (or more) 0-days have to be discovered that the attacker is not willing to come up with the expected costs any more, due to the involved risks. This can be indicated as:

$$\frac{\textit{Exploits Discovered}}{\textit{Time}} \geq \frac{\textit{Number of Exploits} \in f\,(\,r\,(acc))}{\textit{Time}}$$

In other words, 0-day governance has to discover as many or more exploits as the risk calculi of multiple attackers can bear during the same time.

All the force multipliers making life easier for the defender than for the attacker that we indicated earlier go into the calculations at different points. An essential advantage among those is the difference in time needed for exploit discovery on the defender's side compared to exploit design and operation on the attacker's side. Time multiplies the defender's efficiency. Many more exploits can be discovered than designed and operated for a given period of time, assuming the same base of hackers. Other advantages can be highlighted too. Discovery is also less difficult than development, again adding more time and making it easier for the defender to come up with an appropriate workforce — an aspect we will pick up again later on. The defender also has better knowledge about her systems. She can do whitebox discovery, knowing all her system's innards, where the attacker has to do blackbox discovery, prima facie knowing only the outside of the system. If the defender is a government, she might also get her hands on the source code, rendering discovery even easier. And the defender has access privileges that the attacker still has to hijack.

SOME NUMBER CRUNCHING

Some actual number crunching can be done to sketch our idea out in greater detail. But first a note of warning: most of these numbers are educated guesses. They have not yet been measured independently — this is research which remains to be done — and many of them will always be quite individual anyhow. The amount of exploitable code, for instance, will vary greatly depending on the kind of product, its market, and company practices such as quality assurance or external conditions like licensing or the quality of the engineering education in the target's country. But a look at a relevant IT product with some known numbers can provide a first general impression. A major operating system like Microsoft Windows Vista or Apple OS X currently consists of over 80 million lines of code (short: SLOC). The average error rate is not well researched. Studies have found an average of between 3.3 and 8.88 vulnerabilities per 1000 SLOC in some common languages.[25] Cornell once came up with between 1.5 and 5 percent of commercially developed code being faulty, but these are older numbers. Microsoft itself claims an error rate of 0.1 percent, following some serious improvements of their software design process.

If Microsoft is to be believed, this would amount to 80,000 faulty lines of code in Windows. Hackers generally hold that 5 percent of these are exploitable in new ways. Exploits could probably be distinguished by the degree of novelty. For example, hackers could use an old method or a known tool on a new piece of software, amounting to an 0-day which could have been known in

[25] http://www.securitymetrics.org/content/attach/M35Presentations/Doyle-AppMetrics.pdf, slides 8 and 9, accessed 21 February 2012.

principle, or they could come up with an entirely new perspective, fresh from research and combined with intelligence methods in entirely novel ways, amounting to an 0-day in the best possible sense. Some very interesting things happen when hacking and intelligence methods are combined. But because the numbers on all this are coarse and so far just guesses from practitioners, there is no basis for such distinctions at present anyhow. Five percent of 80,000 faulty lines of code amount to about 4,000 possible 0-day exploits. Now how would this amount of insecurity be affected by an effort in 0-day governance? Just as a thought experiment, we could assume that twenty countries join this effort, each with 200 developers. If each developer can discover an average of one 0-day per month (another educated guess from the penetration testing community), the 0-days discovered during half a year — a typical time span for design and operation for most offensive activities — will already amount to 24,000. Greater exclusiveness can be assured a little by assigning the developers different parts of the code to be assessed or different tasks, depending on their expertise. This could cover all of the MS operating systems and a variety of peripheral products. Such an amount of vulnerabilities will seriously affect the strategic mindset of sophisticated cyber-attackers. It would change their risk, cost, and benefit calculi, and render most of the critical targets unattractive. In the mind of any cyber-commander in the world, cyber-weapons will be unreliable. The "deterrence by lack of confidence" we mentioned earlier would be achieved.

Now what about the costs? Would 0-day governance be economically acceptable? Things look good here, too. 0-day governance will be less expensive than any other serious alternative to fight sophisticated attackers. To stick to our example: if the twenty countries we mentioned join this effort with their 200 developers, this could count as sufficiently significant. The costs for 200 developers are high, but tolerable in comparison. If these developers have to be paid industry-equivalent wages (in Europe) of around 10,000 Euros per month, each country will have to invest 12 million Euros during the half-year, give or take a little for extra equipment, know-how, and tools, for the organization of the disclosure process, and for some legal requirements. This is not entirely cheap, to be sure. But it will still be a lot cheaper than switching to a high-security IT environment that will easily cost in the billions, considering the large-scale change of the environment and the follow-up costs due to loss in performance and its compensation.

Some further advantages can be highlighted, but first, we'll mention some counterarguments.

COUNTERARGUMENTS

There are some problems that have to be addressed and assessed if 0-day governance is to be considered a feasible approach. First, the asymmetry between offence and defence is still quite high. On the one side of this asymmetry, a lot more than just one operating system would have to be checked. Other operating systems might be in use, in addition to myriad applications, hardware, and auxiliary informational products running on critical systems and providing other attack vectors, including IT-security products. The frequently criticized monocultures in IT might work as an advantage at this point, especially in critical areas. Many critical areas tend to use the same kinds of IT landscapes. But still, if one end is secured, attackers can opportunistically turn to everything else and still have a lot of options. It is unknown how many options and how good they are. And the loss of a major operating system as an attack vector might already have a significant impact. But there will be other options. Therefore, the selection process has to be made according to preliminary assessments of criticality. Countries engaging in this joint effort have to discern which products are most critical and have to calculate how much would have to be secured in which time. Also, software being extended or upgraded all the time leads to the awkward situation that security continuously declines, despite higher efforts to secure it. There is simply always more insecurity generated by new, sloppily developed products. This could be changed. Depending on the ratios between developed insecurities and discovered vulnerabilities emanating from 0-day governance — this will be an aspect we will highlight briefly further below. But just to make sure, software in critical areas should not be extended any more. On the contrary, it should be slimmed down to the necessary parts, if possible disconnected from everything that is not essential for its operation, and those few parts and minimal configurations should not be altered ever again, except when hardened. Software companies might come up with arguments for why alteration and upgrades/extensions are necessary all the time. But those arguments will most likely be marketing lies (which are a major problem of cyber-security in its own right). A static system is just not a good business case. There might be some arguments from a functional and technical point of view as well. But — given the increased scrutiny due to the security problems involved — those will most likely not be very heavy. A zero-point-something percent rise in efficiency cannot be counted as an argument to jeopardize security.

The vast amount of code to be secured is one side of the asymmetry. On the other side, sophisticated attackers frequently need only a very few 0-days to implement a successful attack. A single 0-day can already be sufficient. This might look like an argument that renders the whole idea futile. With the large and always extended amount of exploitable code on the defender's side, there will always be an 0-day to penetrate critical systems. But this is not a very good argument as such. Available 0-days could become something of a needle in the haystack. The attacker would have to find an exploit that was missed out in the defender's discovery process. And as attackers and defenders function in the same way, proceeding along the same lines in their discovery processes, it is unlikely that the attacker will find that very special exploit. This is open for discussion of course. But two points are worth noting. First, the defenders are not attacking the actual attack development and life cycle, but more narrowly the attacker's strategic mindset. And his strategic mindset has to consider his attack development as rather uncertain and unreliable, even if he might be lucky with his particular exploit. In other words, the attacker might think he stands a good chance. But he cannot know it with certainty. The effect of his loss of confidence is still working. Second, there seem to be implicit hierarchies of vulnerability discovery. Again, this is not verified and is open for research, but it is a common phenomenon that security researchers work on the same vulnerability without knowing of each other. This indicates some kind of intuitive order in the discovery process that will affect the probabilities involved in favour of the discoverer. It's rather likely that the attacker will look for a vulnerability that someone else is looking for too.

Nonetheless, the asymmetry is still striking and will remain a problem. It will have to have an impact on the overall design of the disclosure process.

GETTING THE NERDS TO DO THE JOB

The number of hackers (generally called “penetration testers”) needed for the discovery of all these exploits will be high, and these hackers have to be good at what they are doing. The discovery process certainly demands fewer skills than the overall exploit design and operation – another advantage for the defender. But it is demanding nonetheless. It needs a certain amount of education, of practical experience, and a lot of hands-on work. A certain part of this discovery work can be automated. The penetration-testing community — an unorganized cluster of small and medium businesses scattered around the world — has a lot of tools developed for discovery, which states can obtain. They could be combined in a process of “tool fusion.” This could be highly profitable for the defender again. Recall that professional attack teams already use extensive automation to discover vulnerabilities, reducing the time needed for discovery. One of the reasons for this is that discovery is also the process with the best tool support. This, again, works for the defender. The computer security industry, however, has mostly developed tools and processes to cover existing exploits found in the “wild.” An automated vulnerability detection for the purpose of automated 0-day defence generation is relatively new, but it could yield good automation innovations. It could be researched. Also, a closed and air-gapped cloud or similar supercomputing capacities could be used to enhance the efficiency of tools. But still, a large work force of specialized experts will be needed. And at present, it seems uncertain if the IT world can come up with a sufficient number of skillful hackers. A focused engineering education, however, might alleviate this problem within an acceptable time span and for acceptable costs.

INTEGRITY OF THE DISCOVERERS AND MECHANISMS OF TRUST

In addition to the sheer numbers needed, people also need incentives to engage in such an effort. This might be difficult since they are already paid very high wages in industry. Also, 0-days can always be sold on the black market. Good ones sell for up to two hundred thousand Euros and more. Their mean time to failure will be shortened by a joint effort in 0-day governance, but criminals are quicker than militaries with their operations. Thus, their risk calculi are less affected, and a short-time use of an 0-day might still be of some value, even if the whole process of 0-day governance will have an impact on the exploit black market. This integrity problem could be manageable if the work force is reliable and not criminal. However, the states engaging in this effort themselves might be unreliable. Because offensive cyber capabilities are still attractive, quite a few with the necessary resources to join an alliance on 0-day governance might feel inclined to save the very best 0-days discovered for themselves. Such a behaviour is likely, and mechanisms have to be designed to discourage it. For example, states might (1) install inspectors as an integral part of the discovery teams, (2) have adversary countries check the same stack of code competitively, (3) demand fixed quotas of certain types of 0-days and punish noncompliance, (4) design clear methodologies for what has to be checked in which order to create liability in case knowable vulnerabilities have been intentionally “overlooked” (although such methodologies could still be abused by the attacker), and (5) do sporadic independent assessments of certain parts of their work.

PATCHING

Discovering four thousand 0-day vulnerabilities is one thing. Patching them all is quite another. This has to be done by the engineers inside the companies distributing the specific product assessed. Usually, patching a handful of vulnerabilities takes a few days. Qualys once measured nineteen days for external and sixty-two days for less critical internal systems.[26] Other more recent research (2006) has shown that between 3 and 45 percent of vulnerabilities had been patched within the first thirty days after release while between 0 and 15 percent of vulnerabilities took about 180 days to get patched[27] — for the big software vendors MS and Apple. That already leaves much to be desired as is, so patching even more vulnerabilities will require an appropriately scaled work force. This work force might not be available at present. And its work will not come cheap. Patching can be expensive, especially when it comes to hardware and firmware.[28] But this is rather individual again, and it is quite unlikely that the sum would be as expensive as the high-security alternatives. It should still be cheaper. However, in this case, neither the state nor the client pays, but the vendor. This might discourage vendor cooperation. However, countries are less willing than ever to accept vulnerable products in their critical areas, let alone in their economies. Thus, the losses for companies simply refusing to patch their products could be rather big (notwithstanding the irrationality and irresponsibility of such a refusal). Also, there will be some return on investment as companies get more reliable, stable products in the long run and valuable experience. Also, the discoverers could do part of the patching work as well to alleviate some costs from the vendors. But again, the numbers, the costs, and the needed work force are not known and remain to be assessed. This also holds for the costs to be expected on the side of the client. In the end, vendor-issued patches will have to applied by the clients and this needs some extra time and resources. This is another economic factor to be assessed.

Patching also entails other problems. The whole process of how to diffuse vulnerabilities needs to be organized in a reliable fashion. Responsible disclosure through CERTs seems to be a favourable disclosure policy. Research has shown that immediate and open disclosure through mailing lists such as Bugtraq do not actually provide the benefits their stakeholders assume. Vulnerabilities are not patched faster or more responsibly when publicized—they seem to be patched even slower.[29] Also, good methodologies have to be invented to prioritize incoming reports for how critical the documented vulnerabilities are. And finally, some vulnerabilities cannot be patched at all. If such problems are discovered, the only option to secure the systems from attackers is to isolate the vulnerability. That will need systems engineers who can work closely together with engineers of potential target structures to find individual work-arounds. These are just a few problems associated with the process of patching—more will likely come up. Researching and organizing the patching process will be part and parcel of more detailed concepts of 0-day governance.

[26] See "Laws of Vulnerabilities," in *Qualys Research Report*, http://www.qualys.com/docs/Laws-Report.pdf, accessed 23 February 2012.

[27] See Stefan Frei, Bernard Tellenbach, and Bernard Plattner, "0-Day Patch: Exposing Vendors (In)security Performance," http://www.techzoom.net/publications/0-day-patch/index.en, accessed 23 February 2012.

[28] Hardware companies might in fact be less inclined than software companies to comply to large-scale patching efforts.

[29] See Ashish Arora and Rahul Telang, "Economics of Software Vulnerability Disclosure," *IEEE Security and Privacy*, 3, no. 1 (2005): 20-25.

However, some aid can be provided here as well. First, it could be considered whether full patches are really needed.[30] This would have to be assessed on a case-to-case basis, but in some cases, defenders could also simply widely publish the vulnerabilities and note their characteristics, similar to deriving signatures. Maybe these characteristics could be designed in a way that's sufficient to detect any kind of exploit making use of said vulnerability. This could significantly lessen the costs of patching. Other short-term options could also be researched on bridge times during which vulnerabilities are known, but cannot be patched. "Meandering configurations" or — on the network level — "network configuration randomization" could be aiding solutions here. These are known "moving target" strategies.[31] Systems with known vulnerabilities could simply change their configurations in unforeseeable patterns, shifting around the vulnerabilities, which would, again, significantly raise the risk of an operation getting detected.

These are some first counterarguments. Their weight still needs to be assessed because many of the numbers and causalities indicated are known only intuitively by the actors involved; they have not been researched independently.

BENEFITS

Depending on the outcome of some empirical assessments of the numbers involved and on some of the counterarguments' assessments, the situation might look good or not so good. Either way, 0-day governance will be an important path to consider. It will always provide a number of attractive benefits:

1. Any number of 0-days discovered will affect sophisticated attackers' strategic mindset, their cost-benefit ratios. Thus, fewer and fewer attackers will be willing and capable to undergo cyber operations. This is a classical approach for a security building measure and has a solid and provable value in itself, even if not all malicious actors under any circumstances can be discouraged.

2. As economic considerations play a vital role in the confrontation of sophisticated attackers, 0-day governance provides the benefit of being affordable and generally cheaper than any other serious alternative. The initial costs inasmuch as the follow-up costs are bearable as they do not require a fundamental change of the IT-environment. Even the monocultures can and should be kept as this reduces the amount of code to be governed.

3. 0-day governance will have good long-term effects too. It will get cheaper over time because — after a joint effort over an initial period of five years — most vulnerabilities will simply have been discovered and removed, and because security will finally be faster than the development of new insecurities. The fact that security will outpace the generation of insecurity in this scenario could also lead to a security saturation in this field, rendering the whole problem a temporary one. The experts grown within this process can be diffused to their countries' industries, ensuring a more thorough design process for future products. And in the long run, unsophisticated cyber-attacks will be affected as well. Unsophisticated attackers need vulnerabilities too — so if the process

[30] John Mallery from MIT deserves credit for this intervention.

[31] See Sushil Jajodia, Anup K. Ghosh, Vipin Swarup, Cliff Wang, and X. Sean Wang (eds.) *Moving Target Defense — Creating Asymmetric Uncertainty for Cyber Threats* (Springer Science, Advances in Information Security, 2011).

of vulnerability discovery is closely coupled with good basic protection, overall IT security will profit from this approach as well.

4. The options for fruitful public-private partnerships are good. As 0-day governance would basically do the job the industry failed to do properly, the IT industry could be asked for cooperation either in monetary terms or by lending know-how, source code, education aid, and other things. After all, getting security problems dealt with and educating a better and larger work force capable of dealing with security problems is in industry's interest as well. What is even better for the IT industry is that they do not have to initiate radical changes to their business models to cope with the persistent security problem of sophisticated attackers.

5. Forcing countries to deliver a certain quota of 0-day vulnerabilities will also function as a small, but real and reliable effort on cyber arms control because countries have to put at least a certain amount of their offensive capabilities to this peaceful use, rendering them unavailable for the design of cyber weapons.

6. 0-day governance is beneficial to online privacy. In itself it does not require any degree of Internet surveillance or information control at all. And as it dries out the resources needed for any kind of malicious behaviour in the long run, it also renders online policing less necessary. This holds for any approach by passive security of course, but it is an interesting benefit for European approaches to cyber-security nonetheless, which consider privacy a higher value than most other cultures. As 0-day governance aims merely at generating sufficient passive protection, it is neutral towards privacy-invasive Internet control regimes.

7. The same neutrality will also be of help in generating the necessary enthusiasm among countries to join in on an effort in 0-day governance. Since nothing else needs to be agreed upon and since the security benefits are equally interesting for all, there should not be any diplomatic problem up front.

8. Finally, because the sophisticated attacker could become rather insignificant, current concerns in international relations such as efforts on norms in cyberspace, problems with attribution, and the applicability of international law will be less pressing.

FINAL REMARKS AND OPEN QUESTIONS

Historically, similar approaches have been used by hackers. In the late 1990s, hacker groups would primarily audit programs that would be used by other hacker groups. Two prominent examples are the OpenSSH secure remote shell software and the Apache Web Server used for serving websites. While both tools are widely used by everyone, the hacker groups would develop exploits based on vulnerabilities they discovered primarily to break into other hackers' computers to show their own superiority. This caused many other groups to look into the same programs and to share insights and vulnerabilities with friends and associated groups. There were not too many hackers involved in these cases, but within the time frame of just a few years, both the Apache and the OpenSSH software projects developed into comparably secure software, due to the combined effort of otherwise competing entities. It should, however, be duly noted that both software projects also refrained from excessively adding new features that could introduce new vulnerabilities.

0-day governance should still be accompanied by other efforts to raise cyber-security. Apart from basic protection — consisting of awareness, basic technical protection, basic organizational protection, and regulation to enforce compliance — the basics of all IT-security problems still need improvement, technically, organizationally, and regulatory. At least in all critical areas, high complexity, sloppy development, incompetent and risk-ignorant use, and excessive networking should not be tolerated. A paradigm change needs to be initiated here. High-security IT should become a market in the long run. To enable this market, it should be declared a designated research effort. It will not be as pressing anymore. But it will still be worthwhile. In other words, 0-day governance should not be taken as an excuse for the industry to continue developing vulnerable systems, pushing the costs for higher reliability and security in part to taxpayers.

0-day governance, enforcing basic protection and research in high-security IT, can provide a threefold cure for the cyber-security problem. 0-day governance could be an important part of that triad — especially during the present phase of heavy global investments in offensive cyber capabilities. It could discourage many of these dangerous attackers from entertaining further interest in the matter. Many militaries, secret services, and organized criminals might not feel it's worth the effort anymore. In current times, this can serve as a crucial step to cyber peace. Countries having more to lose than to gain (basically all information societies) might decide to join this kind of solution instead of being drawn into a more expensive and complex offensive cyber rat race.

0-day governance might even be achievable in the near future. Joining states could be obliged to use their offensive work force for this purpose, incentivize researchers accordingly, and hire other personnel from the market at the cost necessary. Basic capabilities could be obtained rather quickly this way. The organizational work to be done is straightforward.

1. First, some basic assessments need to made about (a) products used in critical areas, (b) code complexity and exploitability of these products, (c) exploit discovery rates needed to thwart attackers' cost-benefit calculi, and (d) numbers and kinds of experts needed for exploit discovery. Researchers and industry can do this to provide a basic outset.

After that, the process of 0-day governance needs to be designed. It has to be determined (a) which quotas of which kinds of exploits are to be delivered by which participant (including penalties for non-compliance), (b) how the industry can help by providing source code and know-how, (c) how the process of governing the disclosure and dissemination of results to industry and participants will be designed,[32] (d) how the body to carry out this process is to be designed, and (e) how the process of patching can be organized effectively.

But all this appears to be realizable in the short term, if sufficiently determined regulation sets in.

Some other questions surrounding the economics of 0-day discovery and development will need more research. They have been mentioned already. Most pressing would be:

1. the error rates and the exploitability of these errors of different IT products still have to be determined,
2. the time and costs involved in the different parts of the development- and the life-cycle of exploits have to be clarified, and
3. the risk aversion of sophisticated cyber attackers and the actual impact on their offensive strategic mindset have to assessed.

Some other numbers are needed as well and the math should be refined. But if the basic intuitions of the military offensive and the penetration-testing community are believable — and there is a lot of overlap among those — no contradictions or serious problems should come up. The basic argument made will remain valid. 0-day governance seems sufficiently effective and more affordable than any other real alternative. It's beneficial for every stakeholder in the field. And it serves a number of security goals at the same time. 0-day governance might be a step towards a solution of the cyber-security problem.

Sandro Gaycken is a researcher in technology and security at the Freie Universität Berlin, Institute of Computer Science. He is a graduated philosopher with a doctorate in technology research. The main focus of his research is on IT and society, involving topics like cybersecurity, cyberwarfare, hacking and hackers, surveillance and privacy, open source, hacker ideologies, information society. Apart from these main areas, Sandro also researches the philosophy of security and war, the connex of technology and politics, the nexus of strategy and technology, and the trade-off relationship between security and freedom.

Felix "FX" Lindner is the technical and research lead of Recurity Labs with 18 years computer technology experience, over ten years experience in the computer industry, almost all of them in consulting for large enterprise and telecommunication customers. He possesses a vast knowledge of computer sciences, telecommunications and software development. His background includes managing and participating in a variety of projects with a special emphasis on security planning, implementation, operation and testing using advanced methods in diverse technical environments.

[32] States joining in on 0-day governance should discover their exploits in secrecy and disclose them only in certain intervals to a multilateral organization governing the disclosure process. The disclosure process has to be highly secret too, to further raise the risk of detection for sophisticated attackers.