Next Generation Attacks on the Internet

Position paper for
EU-US Summit Series on Cyber Trust
Workshop on System Dependability & Security,
Dublin, 15-16 Nov. 2006

Evangelos Markatos
Foundation for Research
and Technology Hellas (FORTH-ICS)
Greece
markatos AT ics.forth.gr

Angelos Keromytis
Columbia University
USA
angelos AT cs.columbia.edu

1 Defending Against Next Generation Attacks Through Network/Endpoint Collaboration and Interaction

(Define 1 or 2 difficult problems (research challenges)

Over the past few years we have seen the use of *Internet worms*, *i.e.,* malicious self-replicating programs, as a mechanism to rapidly invade and compromise large numbers of remote computers. Although the first worms released on the Internet were large-scale easy-to-spot massive security incidents [MSB02, MPS$^+$03, SM04, BCJ$^+$05b], also known as *flash worms* [SMPW04], it is currently envisioned that future worms will be increasingly difficult to detect, and will be known as *stealth worms* [SPW02]. This is partly because the motives of the first worm developers were centered around the self gratification brought by the achievement of compromising large numbers of remote computers, the motives of recent worm and malware developers are centered around financial and political gains. Therefore, although recent attackers still want to be able to control a large number of compromised computers, they prefer to compromise these computers as quietly as possible, over a longer period of time, so as not to be detected by any security defenses. Thus, to achieve a stealthy behavior, these attackers have started using, or at least have the capacity to use a wide variety of mechanisms that will make their worms more difficult to detect. Such mechanisms might include:

- **Encryption**. Attackers may communicate with the potential victim using a secure (encrypted) connection, making it difficult for network-based Intrusion Detection Systems [Roe99, XCA$^+$06] to spot their attempted attack.
- **Metamorphism.** The body of worms usually contains some initial code that will be executed when the worm invades the victim computer. Metamorphism obfuscates this code by adding various instructions to it, and/or by substituting blocks of instructions with equivalent blocks of other instructions [SF01]. In this way two “copies” of the worm would be completely different from each other confusing worm detection systems which depend their effectiveness on the fact the all copies of a worm are practically identical [SEVS04, KK04, AAM05].
- **Polymorphism.** Polymorphic approaches obfuscate the worm’s body by encoding it and prepending a decoder. When propagating, the worm mutates its body so that two “copies” of the worm would look completely different from each other (modulo the body of the encoder) [Szö05, DUMU03, K201]. Much like metamorphic approaches, polymorphic systems confuse worm detection systems which depend their effectiveness on the fact the all copies of a worm are practically identical [SEVS04, KK04, AAM05].
- **Hit Lists.** The first versions of recent worms selected their victims completely randomly, *i.e.,* by generating a random IP address in the range `0.0.0.0` to `255.255.255.255`. It has been proposed however, that worms may be more effective if they first create a *hitlist* of all vulnerable computers and then attack only computers in the hitlist [SPW02, AAMA05]. This hithist may even be filtered to exclude honeypots [1]. Armed with a hitlist, a worm is able to compromise a number of vulnerable computers gener-

[1] A honeypot is a computer waiting to be attacked. Once attacked, the honeypot records as much information as possible so that the security administrators will be able to characterize the attack and generate a signature for it.

ating the minimum amount of traffic possible, evading detection mechanisms based on visible traffic anomalies.

- **Hybrid Worms.** Traditional worms used to invade computers by exploiting vulnerabilities of applications listening for Internet connections. However, as more and more computers are hidden behind firewalls and do not listen for incoming Internet connections, they are theoretically protected from such types of attacks. However, to compromise computers protected behind firewalls, worm developers may exploit several different invasion paths including, infected email attachments, infected files shared through peer-to-peer (P2P) networks, and infected files accessed through locally shared disks [KE03]. In this way, an attack may enter an organization as an email attachment, may spread to individual departments through infected disk shares, and may jump from department to department through traditional remote procedure calls.

- **Defense Mapping.** Many of the proposed (and deployed) techniques for detecting and countering new attacks use honeypots as the early warning system [Spi03, DQG$^+$04, YBP04, CBMM04, BCJ$^+$05a, RMT05, MVS01]. However, recent work has shown that attackers can exploit certain features and aspects of a honeypot's behavior to identify and avoid such detectors [BFV05, SII05, RMT06]. Combined with hitlists, this can render worms (especially slow-spreading ones) and other automated attacks virtually undetectable.

- **Client-side Attacks.** In the past year (2005–2006), we have seen an increase in the use of zero-day attacks aimed at client software (especially browsers, but also various types of document viewers such as Microsoft Word, Excel and PowerPoint, and Adobe Acrobat). Other than stand-alone, host-based intrusion detection/prevention mechanisms (such as virus scanners), very little has been done in hardening vulnerable client systems.

1.1 Impact of failing to solve the problem

(For each problem, describe the impact of solving or failing to solve the problem and a description of specific challenges making the problem difficult)

Compromised computers can be used to cause harm to third parties or even to cause harm to their traditional owners.

- **Attacks to third parties.** Recent worm writers organize compromised computers into botnets, *i.e.,* armies of hosts which are primarily used for malicious acts including launching of DOS attacks, blackmailing, sending of SPAM mail, click fraud, theft of intellectual property, and even identity theft. One would envision that botnets in the future could be used for political purposes as well.

- **Attacks to the owners of compromised computers.** A compromised computer can be used to steal the private data and the identity of the owner of the computer. Once, however, ordinary users start to realize the dangers of a compromised computer, they will probably get increasingly less inclined to trust their computers for financial transactions as well as private communications. This will probably *impede the adoption of the information society* and may eventually reduce its overall spread and impact.

1.2 Research Directions

(Provide potential research directions for handling the problem, and metrics for measuring progress against the problem. Possibly, describe also the limitations of current approaches, promising approaches not yet fully explored and desirable approaches where relatively few ideas have been proposed.)

Over the last five years there has been significant research in the area of detection and containment of cyberattacks. Indeed, we believe that we have currently reached the point where it is possible to detect rapidly spreading and massively parallel flash worms. However, it is unclear whether we have the complete technical knowledge or the deployed mechanisms in order to detect and contain *stealth attacks*. Using a combination of the techniques described earlier, such attacks can become invisible (or at least very difficult to detect) to network-based defenses.

Our view is that such attacks can only be detected via large-scale collaboration among end-hosts: by exchanging and correlating relevant information, it is possible to identify slow and stealthy attacks, and to take appropriate measures to defend against them, or at least quarantine those nodes that appear to have been compromised. Specifically, we believe that it is increasingly important to include home and small business computers in the attack-detection process. These computers are increasingly becoming the primary targets of most attackers. Therefore, including them in the worm detection process will increase the chances of worm detection. Exemplifying a large range of access patterns and a large range of applications, these computers typically tend to have more representative configurations than the traditional honeypots currently being used in worm detection. Furthermore, ordinary computers being used by their regular owners are more difficult to be categorized as honeypots and to be avoided by future attacks. The inclusion, however, of home computers in the detection process, should (1) guarantee the safety of the end computer and (2) the minimum possible intrusion in the ordinary use of the computer.

On the other hand, we are not completely discounting network-based defenses: rather, we believe that such defenses must be integrated with end-host defenses. In the past, network and end-host security were viewed as two distinct areas that were meant to complement each other but kept separate. While this allowed for a clean separation between the respective security mechanisms, it also meant that the potential of both was stunted. Furthermore, by keeping them isolated, it was (and is) impossible to exploit scale for defensive purposes. Exploiting scale is something that attackers have learned to do well, as evidenced by such phenomena as distributed denial of service attacks, self-propagating worms and botnets.

The industry is beginning to follow such an approach, albeit in a fragmented, ad hoc fashion. For example, several enterprises exchange alert and IDS logs through sites such as DShield.org; anti-virus vendors with extensive presence on the desktop are correlating information about application behavior from thousands of hosts; network security and monitoring companies perform similar correlation using network traces and distributed blackholes (honeypots). To the extend that such approaches are being explored, they seem largely confined to the realm of information gathering. This also largely seems to be the situation with the DoD and the various agencies. For example, DARPA is currently funding the Application Communities effort, which seeks to leverage large software monocultures to distribute the task of attack monitoring — again, an approach confined to the end-host. Previous work (notably in the DARPA OASIS program) looked into the space of reactive security, but only considered small-scale environments. Arguably, we need to extend the reach of our collaboration-based mechanisms to counter such pervasive threats as DDoS and botnets.

Thus, we argue that it is important to transition into an network architecture design where networks and end-hosts, in various combinations, can elect to collaborate and coordinate their actions and reactions to better protect themselves (and, by implication, the network at large). There are several research issues arising in such an environment, including:

- what problems are best addressed through a collaborative approach
- new mechanisms at all levels of the network architecture (routers, protocols, end-hosts, processes, hardware) that are "collaboration friendly"
- metrics that quantify the security of collaborative approaches over non-collaborative approaches
- who to trust, and to what extend
- how to prevent attacks that exploit such mechanisms, including insider threats
- command-and-control *vs.* loose-coupling mechanism composition

Furthermore, in an era of distributed software services (what is fashionably called "Web 2.0"), no single application, node, or network has enough information to detect and counter high-level semantic attacks, or even some of the more conventional web-based malware (*e.g.,* cross-site scripting attacks). Large-scale distributed systems require large-scale distributed defenses. This is particularly so within specific application domains (such as healthcare and industrial SCADA control), where large-scale collaborative (but independent) defenses will allow better control to critical information and resources.

2 Application-Level Reflection Attacks

(Define 1 or 2 difficult problems (research challenges)

Over the last few years we have seen an increase in the use of botnets, that is, compromised computers being used by attackers for a wide variety of malicious purposes including sending spam email, blackmailing targeted victims, and preventing victim computers from functioning. Although we will continue to see the use of botnets in the near future, we also expect to see the increasing exploitation of *non-compromised computers* for malicious acts. In this scenario, attackers carefully trick non-compromised and possibly non-cooperative computers into acting on behalf of the attackers. For example, in PuppetNets [LAAA06] it was shown that attackers which control a web server may direct a large number of ordinary web clients towards repeatedly requesting web pages from a victim computer. Made possible by the intelligent use of Javascript, these repeated requests where completely transparent to the user in front of the browser, who, all things being equal, did not see any malicious attack being going on. Similarly, Athanasopoulos *et al.* showed that peers participating in the Gnutella file sharing peer-to-peer network can be easily tricked into believing that a victim computer serves a large number of popular files, which in turn, directed a large numbers of requests towards the victim computer, possibly depleting it of its resources [AAM06]. Overnet, another real world P2P system for file sharing, may also be abused in the same fashion [NR06]. Randal Vaughn and Gadi Evron, in a preliminary work, published some techniques to use the DNS [MD88] system as an amplification platform for Denial of Service attacks to third parties by sending malcrafted DNS requests [VE06]. We believe that in the near future we will see an increasing number of such vulnerabilities which will make possible the use of a large number of non-compromised and non-malicious computers into malicious activities.

We call the collective exploitation of these non-malicious clients *Application-Level Reflection Attacks*. As computer applications become increasingly diverse, such attacks may spill beyond the traditional world of computers towards other networks, including, for example, the telephone network. For example, by exploiting a number of Skype (or other Internet telephony) clients, attackers may jam the telephone numbers and/or faxes of victim organizations with bogus telephone calls. To make matters worse, jamming may also be directed to organizations providing vital information services as well, precluding these organizations from providing their service even at critical times. And, since this attack originates outside the traditional telephone network, it might be difficult to trace the attackers back using traditional telecom-style traceback mechanisms. To make matters worse, even when the attack is traced back, the last point in the trace may be a set of non-compromised computers tricked into making these bogus telephone calls. Thus, attackers who have the power to selectively clog a decent number of telephone lines, may use this power for blackmail, revenge, or even to terrorize a selected subset of the population.

2.1 Impact of failing to solve the problem

(For each problem, describe the impact of solving or failing to solve the problem and a description of specific challenges making the problem difficult)

As computers are interwoven within several other services in our society, computer security problems diffuse into the general fabric of the society. Thus, a problem which originally initiated in the cyberspace, may easily transcend into other spaces as well. For example, although SPAM used to be a problem of people communicating via email, it is becoming increasingly easier for SPAMers to make SPAM telephone calls, to send SPAM voice messages, to send SPAM SMSes, to send SPAM MMSes, and in general to SPAM traditional low-bandwidth communication channels using the efficiency, precision, and speed of modern digital computers.

Such attacks can be used for several reasons including:

- Disrupting the operation of organizations who depend on reliable communication channels (*e.g.,* telephones, mail, Internet). Such organizations may range from airline reservation services to the friendly neighboring take out Chinese restaurant.
- Cyber-vandalism against neighborhoods or even towns by selectively clogging their communication channels in a time of need.
- Intimidating, or even terrorizing, large numbers of people by clogging their telephone access to local services such as hospitals, schools, etc.

2.2 Research Directions

(Provide potential research directions for handling the problem, and metrics for measuring progress against the problem. Possibly, describe also the limitations of current approaches, promising approaches not yet fully explored and desirable approaches where relatively few ideas have been proposed.)

The problem of Application-Level Reflection Attacks is relatively new and there exists very little research in this direction. However, as the problem is getting better understood, we envision research along the following lines:

- **Document the extent of the problem.** At the time of this writing it is not clear which systems can be inadvertently used for Application-Level Reflection Attacks.
- **Document the impact of the problem.** We need to develop scenarios which clearly show what is the impact of the problem and what is the financial, social, and political cost associated with it.
- **Develop Detection Mechanisms.** Since Application-Level Reflection Attacks consist of non-compromised computers which behave in a seemingly normal way, we need to develop sophisticated mechanisms for detecting them.
- **Develop Defense Mechanisms.** Once a malicious behavior is detected, we need to have in place defense mechanisms which will block, or at least restrain this type of attack. We envision defense mechanism both close to the host, as well as close to the victim(s).

In closing, we believe that Application-Level Reflection Attacks show that it is technically possible to attackers to perform major attacks against traditional services by manipulating the behavior of a large number of non-compromised computers. Since these attacks involve non-compromised computers, they may be more difficult to detect using traditional approaches, and may deliver a more effective blow, especially when they spread into critical infrastructures, such as the traditional telephone network.

References

[AAM05] P. Akritidis, K. Anagnostakis, and E. P. Markatos. Efficient content-based worm detection. In *Proceedings of the 40th IEEE International Conference on Communications (ICC '05)*, 2005.

[AAM06] Elias Athanasopoulos, Kostas G. Anagnostakis, and Evangelos P. Markatos. Misusing unstructured p2p systems to perform dos attacks: The network that never forgets. In *ACNS*, pages 130–145, 2006.

[AAMA05] S. Antonatos, P. Akritidis, E. P. Markatos, and K. G. Anagnostakis. Defending against hitlist worms using network address space randomization. In *WORM '05: Proceedings of the 2005 ACM workshop on Rapid malcode*, pages 30–40, New York, NY, USA, 2005. ACM Press.

[BCJ$^+$05a] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson. The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In *Proceedings of the 12th ISOC Symposium on Network and Distributed Systems Security (SNDSS)*, pages 167–179, February 2005.

[BCJ$^+$05b] Michael Bailey, Evan Cooke, Farnam Jahanian, David Watson, and Jose Nazario. The blaster worm: Then and now. *IEEE Security and Privacy*, 3(4):26–31, 2005.

[BFV05] J. Bethencourt, J. Franklin, and M. Vernon. Mapping Internet Sensors With Probe Response Attacks. In *Proceedings of the 14th USENIX Security Symposium*, pages 193–208, August 2005.

[CBMM04] E. Cooke, M. Bailey, Z. M. Mao, and D. McPherson. Toward Understanding Distributed Blackhole Placement. In *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, pages 54–64, October 2004.

[DQG$^+$04] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levine, and H. Owen. HoneyStat: Local Worm Detection Using Honepots. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 39–58, October 2004.

[DUMU03] Theo Detristan, Tyll Ulenspiegel, Yann Malcom, and Mynheer Underduk. Polymorphic shellcode engine using spectrum analysis. *Phrack*, 11(61), August 2003.

[K201] K2. ADMmutate, 2001. `http://www.ktwo.ca/ADMmutate-0.8.4.tar.gz`.

[KE03] Darrell M. Kienzle and Matthew C. Elder. Recent worms: a survey and trends. In *WORM '03: Proceedings of the 2003 ACM workshop on Rapid malcode*, pages 1–10, 2003.

[KK04] Hyang-Ah Kim and Brad Karp. Autograph: Toward automated, distributed worm signature detection. In *Proceedings of the 13th USENIX Security Symposium*, pages 271–286, 2004.

[LAAA06] V. T. Lam, S. Antonatos, P. Akritidis, and K. G. Anagnostakis. Puppetnets: Misusing web browsers as a distributed attack infrastructure. In *Proceedings of the 13th ACM Conference on Computers and Communications Security*, 2006.

[MD88] P. Mockapetris and K. J. Dunlap. Development of the domain name system. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 123–133, New York, NY, USA, 1988. ACM Press.

[MPS$^+$03] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the slammer worm. *IEEE Security and Privacy*, 1(4):33–39, 2003.

[MSB02] David Moore, Colleen Shannon, and Jeffery Brown. Code-red: a case study on the spread and victims of an internet worm. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 273–284, 2002.

[MVS01] D. Moore, G. Voelker, and S. Savage. Inferring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Security Symposium*, pages 9–22, August 2001.

[NR06] Naoum Naoumov and Keith Ross. Exploiting p2p systems for ddos attacks. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 47, New York, NY, USA, 2006. ACM Press.

[RMT05] M. A. Rajab, F. Monrose, and A. Terzis. On the Effectiveness of Distributed Worm Monitoring. In *Proceedings of the 14th USENIX Security Symposium*, pages 225–237, August 2005.

[RMT06] M. A. Rajab, F. Monrose, and A. Terzis. Fast and Evasive Attacks: Highlighting the Challenges Ahead. In *Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 206–225, September 2006.

[Roe99] Martin Roesch. Snort: Lightweight intrusion detection for networks. In *LISA*, pages 229–238, 1999.

[SEVS04] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. Automated worm fingerprinting. In *OSDI*, pages 45–60, 2004.

[SF01] Péter Ször and Peter Ferrie. Hunting for metamorphic. In *Proceedings of the Virus Bulletin Conference*, pages 123–144, September 2001.

[SII05] Y. Shinoda, K. Ikai, and M. Itoh. Vulnerabilities of Passive Internet Threat Monitors. In *Proceedings of the* 14^{th} *USENIX Security Symposium*, pages 209–224, August 2005.

[SM04] Colleen Shannon and David Moore. The spread of the witty worm. *IEEE Security and Privacy*, 2(4):46–50, 2004.

[SMPW04] Stuart Staniford, David Moore, Vern Paxson, and Nicholas Weaver. The top speed of flash worms. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 33–42, 2004.

[Spi03] L. Spitzner. *Honeypots: Tracking Hackers.* Addison-Wesley, 2003.

[SPW02] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the Internet in your spare time. In *Proc. 11th USENIX Security Symposium*, San Francisco, CA, August 2002.

[Szö05] Péter Ször. *The Art of Computer Virus Research and Defense.* Addison-Wesley Professional, February 2005.

[VE06] Randal Vaughn and Gadi Evron. DNS Amplification Attacks (Preliminary Release), 2006. `http://www.isotf.org/news/DNS-Amplification-Attacks.pdf`.

[XCA$^+$06] Konstantinos Xinidis, Ioannis Charitakis, Spyros Antonatos, Kostas G. Anagnostakis, and Evangelos P. Markatos. An active splitter architecture for intrusion detection and prevention. *IEEE Trans. Dependable Sec. Comput.*, 3(1):31–44, 2006.

[YBP04] V. Yegneswaran, P. Barford, and D. Plonka. On the Design and Use of Internet Sinks for Network Abuse Monitoring. In *Proceedings of the* 7^{th} *International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 146–165, October 2004.