LEET '10: 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats

April 27, 2010
San Jose, CA

KEYNOTE ADDRESS

- ***Why Don't I (Still) Trust Anything?***
  Jeff Moss, Founder, Black Hat and DEF CON

Summarized by Rik Farrow (rik@usenix.org)

Jeff Moss began by visually describing his presentation style. He showed slides that compared the approaches of Steve Jobs and Bill Gates when making presentations, with the Gates version a confusing (but symmetric) display of dialog boxes. By comparison, Jobs' slide was austere.

Moss was introduced to computers, and shortly thereafter, the world of dialup bulletin boards, in the '80s. He went on to found DEFCON, and later the very profitable Black Hat series of conferences. Currently, Moss is a member of the Homeland Security Advisory Council (HSAC), a volunteer position that he takes quite seriously.

As his first example of why he doesn't trust anything, Moss pointed out that we still don't have working email security. He displayed the headers from an email allegedly secured using TLS, but in reality, TLS was only used by the last forwarding server. PGP has proven to be too hard for most people to use, and also does not have perfect backward security: if you lose your secret key, all of the past email encrypted with that key can be decrypted as well.

Moss then mentioned that he had started keeping his passwords on a "secure" USB fob. Someone bet that he could recover the keys, and by carefully removing the layers from the chips in the fob, was indeed able to read out the bits in storage, despite the manufacturer's attempt at making this difficult to do.

Moss does like some Firefox add-ons, such as NoScript and Certificate Watcher, which led Niels Provos to ask, "Do you trust your add-ons and plug-ins?" Moss had no answer for that, and went on to point out that you have hundreds of certificate authorities but no way to add or delete them from your browser. Then he wondered why HTTPS is not used more, and Nick Weaver answered that HTTPS adds several round trips, each with delays of 300 ms or more.

After making several more points, Moss asked for questions. Fabian Monrose asked what Moss thought about DNSSEC; Moss thought it might be feasible in five years. Weaver commented that DNSSEC might be good for trust anchors, but has a much more limited chain of trust. Moss commented that you could serve your public key via DNSSEC, but soon DNS answers will grow larger than 8k. Michael Bailey asked what might be a solution. Moss answered that vendors have no reason to include good security in their products, with Google being somewhat of an exception. I asked about HSAC, and Moss said the council is supposed to guard against group think in the government. They meet and render opinions publicly.

BOTNETS

Summarized by Chris Kanich (ckanich@cs.ucsd.edu)

- ***Tumbling Down the Rabbit Hole: Exploring the Idiosyncrasies of Botmaster Systems in a Multi-Tier Botnet Infrastructure***
  Chris Nunnery, Greg Sinclair, and Brent ByungHoon Kang, University of North Carolina at Charlotte

Chris Nunnery and his co-authors acquired packet traces and disk images from machines among the two uppermost tiers of the Waledac botnet's command and control (C&C) infrastructure. While the infected hosts which make up the lower tiers of the Waledac botnet are currently well understood, until now the upper tiers of the botnet's management infrastructure were not.

The C&C infrastructure of the Waledac botnet consists of two tiers: an upper tier server (hereafter referred to as a UTS) and usually six second-level servers (referred to as TSLs). The infected hosts of the lower layer consist of spammer nodes which perform the grunt work of sending the botnet's spam payloads, and repeater nodes which forward communication between the upper layer and the spammer nodes. For quite some time, the TSLs were considered purely an obfuscation layer to hide the location and identity of the UTS, however, this work shows that while the TSLs do provide obfuscation for the UTS, the TSL servers provide several other services to the botnet and botmaster. The UTS performs all centralized, autonomous C&C functionality by hosting the zombie binaries and interacting with the spam template provider.

This work exposes several facets of the Waledac botnet's operation. First, an extensive auditing system exists whereby the UTS can audit the operation of the repeater nodes, either by testing the ability to serve a file or resolve a specially coded domain name to the address of another repeater node. In addition to this auditing system, the UTS keeps logs of commands issued through the repeaters as well as Fast Flux domain uptime; Nunnery remarked that the activity of researchers interacting with the botnet would likely have been recorded.

The researchers discovered that the Waledac developers compile the malware binaries using different affiliate IDs so that different groups can propagate and profit from installing Waledac on victim machines. The repacking of these binaries is outsourced to a site called j-roger.com, a process which takes about four seconds. Records recovered by the researchers saw 157 binaries repacked in a two-hour window.

The Waledac botnet employs a differential spamming platform, propagating both "high quality" (HQS) and "low quality" spam (LQS). High quality spam uses stolen SMTP authentication credentials and a SOCKS proxy, either rented from a third party or on a disjoint subset of compromised machines. The TSL machines send HQS: first they grab the spam target list and template from Spamit (a spam affiliate network) via the UTS, then send the email to both the intended recipients and test accounts at various free email providers in order to test the effectiveness of the spam run. In contrast, the LQS bulk email blasts direct from the Spammer nodes are not sent to test accounts, but successful delivery to the MTA is recorded.

Asked what packer Waledac uses, Nunnery said originally the UPX packer was used, but eventually they moved to a custom repacker. When asked about indications of accounting or cost differentiation/billing statistics, no direct evidence has been found, but there are some figures related to price structure outlined in a file recovered from the file system image. What was the reliability of these machines, and could the botmaster move C&C to EC2? The machines were rented from LeaseWeb and had 99% uptime. Had the botmasters noticed the research group? They certainly had, and there was direct contact between the two parties.

- ***Insights from the Inside: A View of Botnet Management from Infiltration***
  Chia Yuan Cho, University of California, Berkeley; Juan Caballero, Carnegie Mellon University and University of California, Berkeley; Chris Grier, University of California, Berkeley; Vern Paxson, ICSI and University of California, Berkeley; Dawn Song, University of California, Berkeley

Chia Yuan Cho explained the spamming perpetrated by the MegaD botnet, which at its peak was responsible for one-third of all spam and currently sends 15% of worldwide spam. The botnet survived a takedown attempt in November of 2009. MegaD malware binaries have their command and control (C&C) master server address hardcoded, and upon bootup the master server will inform the bot of its template server from which to acquire spam templates, a drop server for binary updates, and an SMTP server for spam testing.

As different binaries connect to different master servers, this raises the question of finding binaries belonging to different master servers. The researchers used a Google hack to find different master servers and “milked” subsequent binary updates using C&C emulation. As the master servers use TCP ports 80 and 443 and imitate a Web server when responding to a simple Web client request, a Google query can be formed based on the C&C server’s response in order to find new C&C servers. Once a new C&C server has been identified, knowledge of the MegaD C&C protocol allowed construction of benign programs which could request new versions of the MegaD malware binary.

After the November 2009 takedown, only one template server survived, and the templates it served stayed the same for a week. After that week, templates pointed to a new test SMTP server; 16 days after the takedown, spam was again being delivered at the full pre-takedown rate. Two possibilities exist as to how this was possible: either reanimation of servers redirected the currently active bots to new C&C servers, or fresh installs repopulated the spamming tier of the MegaD botnet. The former can be ruled out as FireEye (the company that initiated the takedown) reported that none of the C&C servers were still available post-takedown. The latter remains the sole likely possibility, especially as MegaD is known to use generic downloader malware to propagate via a Pay Per Install model.

Another large facet of this work identified two different master server groups within the greater MegaD population. The researchers’ hypothesis that these groups are operated by two different botmasters is supported by many facets of the operation of these disjoint subsets. While each group contains several master servers, one group would use different subsets of the template elements available within the spamming engine; the templates themselves displayed different types of polymorphism and greatly differed in how often the template was updated on the server side. One group used only a Viagra-themed spam campaign, while the other group used a diverse set of spam campaigns not limited to pharmaceuticals. This work was supported by four months of infiltration of the botnet, consisting of milking both the C&C and template servers, Google hacking to find additional servers, and discovering the existence of two disjoint and differently managed botnets under the name MegaD.

Cho was asked why, since C&C servers were hardcoded into MegaD binaries, couldn’t simple C&C server takedowns neuter the botnet. Vern Paxson responded that the botmaster’s counter-solution is just to use the Pay Per Install model to repopulate his spamming botnet cheaply and effectively. Niels Provos asked what the minimum number of bots would be to run a successful spam campaign, and although there was no answer, the data point of 20,000 bots within the Storm botnet as an upper bound was brought up. How long does it take to build up a spamming botnet from scratch? Since 16 days was enough to return to full speed, how does one actually hurt the spammers’ bottom lines? One suggestion was to clamp down on domain registration or find other ways to hurt them monetarily on the back end. Fabian Monrose asked if there are any positive takeaways from this work, and while no direct advantages could be enumerated, the suggestion was raised that concentrating on the economics of the spamming trade might prove fruitful. Can one use templates themselves to infer botnet membership? The answer was yes; work by Pitsillidis et al. on botnet Judo was brought up as an example of a similar strategy based on this intuition.

THREAT MEASUREMENT AND CHARACTERIZATION

Summarized by Srinivas Krishnan (krishnan@cs.unc.edu)

- ***The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution***
  Moheeb Abu Rajab, Lucas Ballard, Panayiotis Mavrommatis, Niels Provos, and Xin Zhao, Google Inc.

Moheeb Rajab explained that the focus on Web malware at Google has been on delivery mechanisms. Since drive-by downloads have been getting more and more attention, malware distribution networks have been looking at other delivery mechanisms, with fake antivirus as an example of this evolution. The attack is simple: an HTML pop-up appears on the user’s screen with an image of an antivirus engine scanning the user’s system. The user is then informed he has malware on his machine and to remove it they need to download the “antivirus” software by clicking on a link. Once the user clicks on the link, the malware is downloaded and installed on the system. The malware then acts as an annoyance, a keylogger or as ransomware, asking the user to pay for the removal of the malware from their system.

Rajab said that this attack plays on the user's fear and allows the attacker to directly monetize the malware. The statistics breakdown shows over 35 million downloads a month and Google's malware tracking system reveals a 3% increase over the past year. Furthermore, fake AV accounts for approximately 15% of Web malware. The authors used the anti-malware infrastructure they built at Google which analyzed 240 million URLs with a 20% sampling rate. The authors also reported that 11,500 domains were involved in the fake AV distribution over the past year, where the rate of increase grew from 90 to 600 domains per week.

The attackers attract users by commonly used techniques: Web and email spam, and exploit ads. There was also a new increase in event-driven exploits based on Google trend keywords; over 70% of malware domains that used the event-driven exploit were also fake AV distributors. The median lifetime of the domain decreased from 1000 hours to little less than an hour, in response to Google's better detection time.

Abhishek (McAfee) asked about how Google protects users: are the domain names reported back in search results? Rajab responded that the results are tagged; users are protected as Google's anti-malware system's detection time goes down. Vern Paxson (ICSI, UC Berkeley) wondered about the economic aspect of the fake AV attacks. Rajab said that they did not look at it, and co-author Niels Provos said that the money trail ends at the bank, so it's hard to get an actual idea of the scale of the fake AV economy. Roberto Perdisci (Georgia Tech) pointed out that fake AVs are persistent; what is the behavior after they are installed on the system? Rajab said they did not do the binary analysis. Jack Stokes (Microsoft Research) suggested that analysis might be improved by involving users. Rajab said that, currently, users can report suspicious URLs, but there is no direct way to tag a URL. Niels Provos pointed out that user input is noisy and poor, based on experience, but further investigation is warranted.

- ***Spying the World from Your Laptop: Identifying and Profiling Content Providers and Big Downloaders in BitTorrent***
  Stevens Le Blond, Arnaud Legout, Fabrice Lefessant, Walid Dabbous, and Mohamed Ali Kaafar, I.N.R.I.A, France

Stevens Le Blond presented the work they did on quantifying user contribution in BitTorrent and if it was possible to identify the content providers. The authors looked at injection and download rate at a large torrent index site (Pirate Bay). They would connect every minute and get the recent uploads and track the respective torrents. They also checked who the first few uploaders for each torrent were and collected their IDs. Using this method, they were able to identify 70% of the content providers. Furthermore, they studied 10,000 IP addresses and classified them as either middleboxes, spies, or monitors. The key insight was that most of the content in torrent networks is contributed by a few content providers and that there is a constant monitoring and spying presence on these networks.

Niels Provos asked if identification accuracy could be based on multiple parties using the same techniques to hide, leading to possible skewing of the results. Le Blond replied that cross-verification of the results shows their results to be 99.9% accurate. Eric Ziegast asked if they saw any differences between music, movies, and code? Le Blond responded that they saw no direct correlation between contents and size. The biggest provider was Easy TV, injecting six new TV shows every day, roughly 500 MBs per TV show. Those providers are using the same machine. We have looked at type of content to see what people were injecting. Ziegast then asked if they saw much malware, and Le Blond said no. Provos asked if Pirate Bay attempts to block peers that appear to be spying, and Le Blond said they do, but you can spy for 50 days before they notice. Provos said that is horrible detection performance.

- ***WebCop: Locating Neighborhoods of Malware on the Web***
  Jack W. Stokes, Microsoft Research; Reid Andersen, Christian Seifert, and Kumar Chellapilla, Microsoft Search

Jack Stokes summarized the work Microsoft has been doing in malicious page detection. The approach is different from other Web malware detectors, as the researchers use a bottom-up approach. The detection engine directly goes to the malware site and then uses a Web crawler to build a Web graph based on the links on the site. The Web graph is used to find the intersection between search results and the links on the malware site by overlaying the topologies. This approach is based on targeted detection of the malware found on a user's system, taking advantage of Microsoft's Anti-Malware (AM) tool.

Vern Paxson asked how you know if a URL is infected. Stokes responded that a human is analyzing and labeling it. Paxson asked how telemetry finds things that crawling does not. Crawling cannot tell if it's malware, but WebCop can tell. (There is room for improvement here, as WebCop does not download binaries and run instances in VM, so cannot tell if malware is real or not.) Fabian Monrose asked if every file that a user downloads is reported to Microsoft. Stokes replied, yes, if AM reports it as unsigned or it has a bad hash. Monrose said he was surprised, and he wondered if users know about this. Stokes said if it's free, you cannot opt out. If you pay money for it, you can opt out. It's in the privacy statement. By this point he was yelling and the room was getting worked up. Stefan Savage pointed out that this practice of collecting data from AV is truly industry-wide. Honeypots no longer work. The only way to deal with this is to use clients such as your sensornet. Monrose asked if we have a responsibility as a community to report this. Jeff Williams, the head of the WebCop group, said that they don't collect personally identifiable information or save the IP addresses.

INVITED TALK

- ***Naked Avatars and Other Cautionary Tales About MMORPG Password Stealers***
  Jeff Williams, Microsoft Malware Protection Center

Summarized by Rik Farrow

Williams said that working at the Malware Protection Center provides a unique insight through protecting one hundred million Hotmail users. His group also pushes out the monthly update to anti-malware (AM). "We get to see where in the world things are, where they come from (Web, email, etc.)."

In this talk his focus was on password stealers. Williams pointed out that, contrary to what you might think, big targets are games and game discussion sites. A game password provides access to avatars as well as anything they have won or collected, and there is a thriving black market in gaming plunder. Also, law enforcement has little interest in protecting people from threats within games, even though the gaming market is on the verge of overtaking film in revenue.

When Williams mentioned that Brazil has a very high number of password stealers, Niels Provos asked why this might be so. Williams said there was no particular reason why, and Provos pointed out that there is no native AV industry in Brazil. Another person mentioned that liability laws are different there as well. Williams agreed, and quipped that Brazilians do more online banking than most other peoples—they just don't use their own accounts.

Williams described eight families of malware used to steal gaming passwords, some of which focus specifically on gaming kiosks. He mentioned the Dogrobot malware which led to kiosk owners wiping and re-installing machines every night to protect customers, and their business.

Williams then laid out the gaming black market:

Envelopes—stolen account info

Stalls—online space for collecting/selling information

Trojans—trojans and malware can be made to order, even online

Trojan generators—software that customizes trojans

Williams finally approached the title of his talk, showing a girl in a bathtub. Following a related link led to the installation of Agent.ABHN and Alureon.BJ. Williams said that in April 2009, there were 860,000 sites containing malicious scripts, leading to exploits against IE, Windows, and third-party ActiveX controls. Currently, Taterf is the most prevalent malware, designed to capture multiple game passwords. Taterf often includes other exploits and malware.

Protecting games is difficult because gamers focus on performance, and AV hurts performance. Williams does something like what Blizzard (World of Warcraft) has done: distributing and encouraging the use of hardware tokens for authentication.

Williams ended with a call to action: attackers have community, defenders should too to advance the state of community-based defense. He pointed to the Conficker working group as an example of a strong model, with cross-industry participation including AV, ISVs registrars, researchers, law enforcement, and many others. Williams said that software has bugs and traditional quality assurance doesn't find them, something I certainly have no trouble believing some eight years after Bill Gates announced "trustworthy computing."

Nick Weaver pointed out that hardware tokens don't work for banking (as malware can use a proxy to abuse authentication), but supposed that this wouldn't be true for gaming because it takes time to sell the loot. Williams countered that shorter timeslices have occurred in gaming crime as well. Stefan Savage wondered if they had dug deeply into the monetization side: where is the value coming from? Had they talked to game designers about reversal of loot loss? Williams said they hadn't, but it would be terrific if this could be done. Someone asked if they had a breakdown of the different types of accounts stolen, and Williams said that they didn't have hard statistics, but that it was not uncommon that a gaming password would work for banking as well.

THREAT DETECTION AND MITIGATION

Summarized by Chris Kanich (ckanich@cs.ucsd.edu)

- ***On the Potential of Proactive Domain Blacklisting***
  Mark Felegyhazi and Christian Kreibich, International Computer Science Institute; Vern Paxson, International Computer Science Institute and University of California, Berkeley

Mark Felegyhazi began his talk with a graph from previous work showing the time of register and time of first use for domain names used in botnet spam campaigns, noting that most spam domains were registered in large batches on the same day. Using the intuition that the scale of the spammers' operation requires batch registration of domain names, this work aims to proactively blacklist domains when they have likely been registered by a spammer before most of the batch has ever been seen in spam. The authors cluster using properties of the nameserver architecture and registration information from WHOIS records. They used two properties of the DNS architecture: whether a domain was self-resolving (i.e., the NS for example.com is hosted within *.example.com) and the freshness of the NS registration (in this case, whether the domain has been registered within the past year). Features including registration date were mined from WHOIS records.

To evaluate their strategy, the authors seeded a set of known "bad" domains from the joewein.net blacklist and tripled the number of known bad domain names via inference. Approximately 75% of the additional domain names eventually end up in various blacklists. Of the remaining unknown domains, visual inspection of many of the pages they host verify their content as spam or malicious, and some remain unknown, as they were not used at all over the course of this study.

What was the runtime of the clustering algorithm? It currently runs in about half an hour but has not been optimized for operational deployment. The clustering requires searching the zone file for domains whenever the blacklist is updated, and this would be very possible in practice. Yinglian Xie of Microsoft Research asked whether the domain names point at the same IP addresses and whether this might factor into clustering. Felegyhazi did not use the domain names' resolution in his clustering. Brent Hoon-Kang of UNC Charlotte asked whether NXDOMAIN results were part of the false positives and whether later domain testing might improve accuracy. Felegyhazi said that this might help, but the existing strategies identify domains sufficiently for blacklisting purposes. What were the characteristics of the unknown domains? They were mostly in the form "nounnoun.com" (as seen in many other spamming campaigns). Fang Yu of MSR asked about the breakdown of registrars for the bad domains. Felegyhazi remarked that this was an interesting point; a table exists in the paper showing a pronounced difference in distribution of registrars between all domain names and blacklisted domains.

- ***Detection of Spam Hosts and Spam Bots Using Network Flow Traffic Modeling***
  Willa K. Ehrlich, Anestis Karasaridis, Danielle Liu, and David Hoeflin, AT&T Labs

Anestis Karasaridis remotely presented this work on spambot detection. The authors' system mainly uses netflow data, along with some DNS data. The statistical properties of the spam at the network level are markedly different from legitimate email in terms of mean and variance of bytes per flow, which allows for network monitoring where the content of the message remains private. The authors identify both spamming hosts and spam command and control machines via their algorithm. While the netflow data is sufficient to classify spamming SMTP clients, the detection of controller machines is a two-stage process. Flow statistics such as fan-in and bytes-per-flow identify possible controllers, and then the second stage aggregates these records and ranks them by number of clients. DNS information is used to detect transient "Fast Flux" domains in order to improve controller detection. Direct connections within the netflow data without preceding DNS lookups also indicate compromised machine access. The authors found controllers for the Zeus, Ozdok, and Cutwail botnets via their approach.

Yinglian Xie asked whether attackers cognizant of these methods could introduce variance to hide from the algorithm, and Karasaridis noted that this was certainly possible, but they had not seen this happen in practice. Xie also asked what would make this spamming approach economically infeasible, because it is not very hard for a spammer to change hosts. The authors did not address economics in their research.

- ***Extending Black Domain Name List by Using Co-occurrence Relation between DNS Queries***
  Kazumichi Sato and Keisuke Ishibashi, NTT Information Sharing Platform Laboratories, NTT Corporation; Tsuyoshi Toyono, Internet Multifeed Co.; Nobuhisa Miyake, NTT Information Sharing Platform Laboratories, NTT Corporation

Kazumichi Sato presented his group's approach to extending blacklists with an aim very similar to that of Felegyhazi's work but with an orthogonal approach. The intuition underlying this work is that, as botnets must plan for resiliency against takedown, bots will often perform lookups for several command and control servers at the same time. The system assumes that if an individual host resolves two domain names at the same time frequently and one is bad, then the other is likely bad as well. A scoring system using a co-occurrence relation is used.

A difficulty with this approach is that popular, legitimate domain names are resolved by bots for either connectivity tests or simply by the legitimate user of a compromised machine. A larger population of non-infected hosts and their lookup statistics are used to weight the legitimate and popular domain names away from being labeled as bad. In addition, a host's overall lookup rate is used to weight queries so that individual heavy hitters do not bias the co-occurrence scoring. The dataset used includes one hour of DNS traffic from February of 2009. The authors created a blacklist using honeypots, including 270 domain names, and found 91% of bad domains within this trace were among the top 1% of all scores; among the top 100 scores, 39 were black, 4 were legitimate, and 56 unsure. The unsure domains included oddities such as lookups of the form <black domain name>.<legitimate domain name>, as well as DNS blacklist lookups.

Eric Ziegast asked whether this system had been deployed operationally since compiling the one hour of data used for the study? Sato answered no, stating that the timestamp mismatch between the blacklists and DNS data prevents deployment.

NEW THREATS AND CHALLENGES

Summarized by Rik Farrow (rik@usenix.org)

- ***Are Text-Only Data Formats Safe? Or, Use This LaTeX Class File to Pwn Your Computer***
  Stephen Checkoway and Hovav Shacham, University of California, San Diego; Eric Rescorla, RTFM, Inc.

Stephen Checkoway pointed out that most people consider text to be safe from infecting their computers. He presented a simple chart with two columns: unsafe and mostly trusted. Under "unsafe," he listed executables, Web apps, media (Flash, JPG, etc.), and Office formats as examples. Under “mostly safe” he had ASCII text.

TeX and LaTeX have interpreters that create boxes with text or equations in them and then glue the boxes together, creating a laid-out page. By convention, the backslash is the escape character that tells the interpreter to do something special with the following text. You can also read and write files, although the *nix versions limit writing files to the current directory.

The authors wrote a virus, sploit.js, that finds .tex files and infects them. Interpreting infected files spreads the virus. They tested some online sites that offer TeX interpretation and found that they were potentially vulnerable to infection. And filtering out reads/writes is not trivial, as TeX includes macros that can obfuscate any virus, as well as a mechanism for changing the escape character, so potentially any string of text can be part of a virus.

Checkoway concluded by remarking that using binary vs text was not a good way to classify uploaded or downloaded files as safe. Niels Provos wondered how many academic sites were used to propagate sploit.js, and Checkoway assured us that they notified sites that provide LaTeX rendering services of the potential for remote code execution.

- ***DNS Prefetching and Its Privacy Implications: When Good Things Go Bad***
  Srinivas Krishnan and Fabian Monrose, University of North Carolina at Chapel Hill

Srinivas Krishnan pointed out that browser vendors poach market share by creating browsers that display pages more speedily. One of the tricks used involves prefetching DNS entries by looking up all the domain names found in links on the current page. The authors examined two ways of determining if a particular Web page had been viewed using two methods: probing open DNS servers and examining a DNS server dump.

They start by creating a profile of domain names that can be used to identify a particular Web page. Then they crawl a DNS server that allows queries from anywhere and serves the network of interest to infer cache hits using cache snooping. As the server replies, it will include TTL values for each domain, allowing the calculation of a confidence rating. This works because they already know the maximum TTL values by querying the domain itself, and if the domains were all fetched within a short time window on the crawled server, this indicates that they are related to a particular set of queries. They discovered that they could scan about every 30 minutes and get good results. Searching a DNS server dump provides a lot more accuracy.

Someone asked just how important this is. If you can dump someone’s DNS server, you could likely just monitor the network and determine which URLs someone is fetching. Krishnan concurred, but pointed out that using logs works best and only works with targeted searches using profiles. Niels Provos wondered about the need for tokenizing, and Krishnan explained that tokenizing is only done when searching logs, not when trawling servers. Fabian Monrose pointed out that prefetching provides a lot more information. Roberto Perdisci asked if they made recursive requests, and Krishnan said they did not. Perdisci then wondered if this would indicate that something is awry. Krishnan responded that most open server operators don’t check and don’t care. Eric Ziegast asked if when doing online probes they can only infer that someone is doing a search, and Krishnan said this is correct. Only DNS server dumps reveal the source address of requestors. Monrose said that Chrome makes it easy to disable prefetching, and I discovered that it’s not that hard to do in Firefox either.

- ***Honeybot, Your Man in the Middle for Automated Social Engineering***
  Tobias Lauinger, Veikko Pankakoski, Davide Balzarotti, and Engin Kirda, EURECOM, Sophia-Antipolis, France

Tobias Lauinger described a man-in-the-middle (MITM) attack that can improve upon IM spamming. IM spamming by bots is easy to detect, with 80% of users detecting a spambot within three messages (Huber et al.). The authors decided to take a different route by using a MITM approach to connect two real users and allowing them to chat until it was time to insert the spam.

Someone asked if they had asked their Institutional Review Board (IRB), and Lauinger said they don’t have an IRB, so there is no one to say no. This provoked laughter from the audience. They did ask the University of Vienna and got approval from them.

They chose to use a dating channel and filtered out initial links. They also used a filter that translated male into female terms, and vice versa, and could keep people chatting up to five minutes. The MITM relay would start inserting URLs after ten exchanges. Using Tiny URLs worked best for this type of attack.

Vern Paxson asked how many sessions were shown in the graph (in the paper and as a slide), and Lauinger answered more than 1000 but not a million. Bill Simpson asked what name they chose, and Lauinger answered that they chose a name that could be male or female.

WORK-IN-PROGRESS REPORTS

The LEET ’10 workshop ended with a short WiPs section.

Nick Weaver (ICSI) had a rant about making work harder for virus writers by suggesting that AV vendors should use polymorphic recovery tools, making them more difficult to disable or avoid.

Stefan Savage (UCSD) wants better ways of predicting the exploitability of things to help sysadmins choose which vulnerabilities are the most important ones to patch. The Common Vulnerability Scoring System (CVSS) is the industry standard today, and it uses a funky equation and several variables filled in by guesswork. So far, Savage’s group has used a metric that employs exploit data from OSDVB, measures the time from first announcement to first exploit, and has a more successful metric than CVSS.

Brent Hoon (Brent ByungHoon Kang, UNC at Charlotte) said that although Conficker had been contained (by buying up the domain names to be used in C&C) in 2009, Conficker has never disappeared. They are still seeing hits, and there appear to be about six million IP addresses that show signs of infection. Stefan Savage wondered how you could economically get patches to six million people. Someone else asked how Conficker was contained, and Hoon said that Microsoft had bought up 10,000 domain names, enough for three years. Hoon also said that Waledac, another contained botnet, has been decreasing over time. Savage commented that we have a low threshold for success.

Working with his advisor, Stefan Savage, Chris Kanich (UCSD) created a site that sat between those wanting CAPTCHAs solved and those willing to do so. They purchased and sold CAPTCHA solving, finding that CAPTCHAs cost .50 per 1000 solved, that it takes 8–12 seconds to solve each one, and where the solvers are. A paper related to this research will appear at USENIX Security '10. Niels Provos asked if there was an IRB review, and Savage answered that they can't identify any of the participants, so there was no review.