LEET '11: 4th USENIX Workshop on Large-Scale Exploits and Emergent Threats

Boston, MA
March 29, 2011

Invited Talk

Summarized by He Liu (h8liu@cs.ucsd.edu)

Tor and Circumvention: Lessons Learned

Nick Mathewson, The Tor Project

Nick Mathewson introduced the Tor project. Tor is free open source software that enables Internet online anonymity, used by approximately 250,000 users every day. Anonymity means an attacker cannot learn who is communicating with whom in a communication network, and Tor aims to be a usable, deployable software that maximizes online anonymity. Anonymity serves different interests for different users. Anonymity is privacy for private citizens, network security for businesses, traffic-analysis resistance for governments, and reachability for human rights activists. To be anonymous, it is important to have company and hide in an anonymity network. The Tor project benefits good people more, because bad people are already doing well. Tor uses a multiple-hop relay network design.

Nick talked about how Tor evolves to have more blocking-resistance, i.e., is harder to be blocked by firewalls. Firewalls can block Tor if Tor's traffic has a unique fingerprint. To defend against these attacks, Tor's TLS handshake mimics the ones between Firefox and Apache. Attackers can also block Tor by preventing users from discovering relays by blocking relay directories or all relay nodes. To defend against this, Tor uses "bridge" relays. Bridge relays are relay nodes that are not listed in any directory, and Tor partially distributes addresses of bridges to users. Bridge relays help users to bootstrap and connect to other relays in the network.

Nick also showed several figures of user statistics of Chinese, Tunisian, Egyptian, Libyan, and Saudi users. These figures show that use of Tor has a strong correlation with big political events. Some of them also showed how Tor's blocking-resistance upgrades are reflected on user counts.

Finally, Nick talked about several challenges for the Tor Project, including how to defend against application-level attacks, how to guarantee Tor software integrity on software distribution, and how to educate people about using online anonymity software for better Internet security.

Dan Geer (InQTel) asked about transition from IPv4 to IPv6. Nick said Tor is now IPv4 only, but IPv6 is under consideration. IPv6 is not universally deployed over the Internet, and hence it is hard to use it to build a world-wide anonymity network. Stefan Savage asked how to debug a blocked node. Nick answered that mostly they manually cooperate with users in different countries to debug. Is Tor mostly used in countries with censorship? Nick said no, that the two biggest user bases are in the US and Germany. Dan Geer asked about defending against traffic correlation attacks. Nick said it is still unacceptably expensive and would require sacrificing too much usability, but they are happy to use it if related research produces more feasible results.

Attacking Large-Scale, Real-World Systems

Summarized by Gianluca Stringhini (gianluca@cs.ucsb.edu)

Exposing the Lack of Privacy in File Hosting Services

Nick Nikiforakis, DistriNet, Katholieke Universiteit Leuven; Marco Balduzzi, Institute Eurecom; Steven Van Acker and Wouter Joosen, DistriNet, Katholieke Universiteit Leuven; Davide Balzarotti, Institute Eurecom

File hosting services are becoming a popular way for users to share files on the Internet. Anytime a user uploads a file to a file hosting service (FHS), the service assigns a URI to it that univocally identifies it and allows other people to download it. Unlike traditional peer-to-peer, they allow people to share a file in a "private" way, since only those who know the URI can access the file.

This work studies to what extent FHSes guarantee privacy, and whether it is possible for an attacker to guess the URI that identifies a file and, thus, illegally download it. Nick said he studied how 100 FHSes generate their URIs, finding out that they typically generate them in a sequential way. In particular, 20 of them do not add any non-guessable information at all, making it trivial for an attacker to enumerate all the files uploaded to the service and download them.

Those services that do not use sequential numbers might appear to have better security, but Nick showed that many of them use short identifiers that are easy to brute-force. After getting the URI that identifies a file, an attacker can find out whether the file has been uploaded as private. To do this, it is enough to search for the file on a search engine. If no results are returned, there is a good chance that the file is private. Nick said that 54% of the files they crawled were private.

Nick said they then created some honey files that were "calling home" when opened, to see whether people crawled for them and downloaded them. These files were downloaded 275 times, by more than 80 unique IP addresses. This is the proof that there is a problem of people looking for private files on FHSes. To check whether the intentions of these people are malicious, Nick created a fake credit card trading Web site and put information about it, and the credentials to log into it, in some of the honey files. In total, they logged 93 logins in the fake Web site, by 43 different IP addresses. This proves that attackers use the information found in illegally retrieved data.

Someone asked if they were blacklisted during their trolling, and Nick replied, Only when they were greedy. They slowed down the rate and were fine. Chris Kruegel asked if any of the top five FHSes were using sequential identifiers, and Nick said that when they reported this issue to one of the top five, they explained they would slow down the ability to search, but not fix sequential numbering in URIs. Someone wondered what was displayed to the users who logged into the fake site. Nick answered that the only thing that was displayed to them was a "come back later" message.

One Bad Apple Spoils the Bunch: Exploiting P2P Applications to Trace and Profile Tor Users

Stevens Le Blond, Pere Manils, Abdelberi Chaabane, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous, I.N.R.I.A., France

Stevens briefly introduced how Tor and the whole onion routing concept works, and then started talking about two attacks they developed against users using BitTorrent trackers through Tor. These attacks nicely complement the keynote by Nick Mathewson, who acknowledged that the Tor team knows about a few privacy issues that involve Tor and peer-to-peer applications. The attacks developed by Stevens and his colleagues require the attacker to control one or more exit nodes in the Tor network. This type of node is really valuable, since it allows the attacker to see the outgoing traffic in the clear, although without any notion about the IP that generated it. The first attack hijacks the list of BitTorrent peers returned by a tracker to include a host controlled by an attacker. Since 70% of the people using BitTorrent over Tor do that only for accessing the tracker, they will connect to the malicious peer directly, disclosing their real IP address. The authors were able to track 10,000 IP addresses using this technique.

The second attack exploits BitTorrent distributed hash tables, which are a way for users to hide which files they downloaded. DHTs work over UDP, which Tor does not support. Therefore, when a user tries to use them through Tor, he will fail, and connect directly to them, publishing his real IP address in the DHT. At this point, by looking at the BitTorrent subscription identifier stored in BitTorrent subscription to the central tracker, it is possible to deanonymize the user who passed through the rogue Tor exit node. Stevens also showed how, after having assigned a real IP to a BitTorrent connection passing through a Tor circuit, it is trivial to identify all the other connections by that host, since they will all pass through the same circuit at a certain time. This attack also allowed them to deanonymize not only BitTorrent traffic but other kinds of traffic such as HTTP. Stevens next showed the breakdown of the searched words on BitTorrent over the exit nodes controlled by them. The vast majority of the query strings were adult content–related.

Chris Kruegel asked how you could do anonymous file sharing over Tor. Stevens replied that this is a fundamental problem with Tor, and the only safe way to do file sharing would be within your own ISP's networks. Someone asked whether using Tor over BitTorrent would be incredibly slow. Stevens replied that the vast majority of the tracked users use Tor only for the tracker data, which is what is usually checked for copyright infringement, but do not use it for actually downloading files.

Studying Cyber-Criminals and Their Tools

Summarized by Rik Farrow (rik@usenix.org)

The Nuts and Bolts of a Forum Spam Automator

Youngsang Shin, Minaxi Gupta, and Steven Myers, Indiana University, Bloomington

Youngsang Shin explained how they had worked with a demo version of XRumer to see how it works. They also compared its features to other forum spam automators in their paper. XRumer can register with forums and automatically deals with the obstacles designed to prevent this. It can solve some classes of CAPTCHAs with built-in algorithms. XRumer can also set up Gmail accounts to be used for receiving an activation email, then interpret the email and send the proper response to activate a forum account. XRumer includes a macro language for customizing spam postings, and the commercial version includes a search tool for finding the appropriate forums to spam.

Youngsang pointed out that XRumer does behave in ways that make it easy to notice. Although it will use different User-Agent tags, XRumer always sends the same HTTP header (and uses HTTP/1.0), and always sends a cookie even if none is required. XRumer also sends a Host header with a hostname but without the port number usually found in IE headers. XRumer also adds a Proxy-Connection header, something that is not a standard and is rarely seen. XRumer is designed to work with proxies to hide the sender's IP address, and proxies typically change the Accept-Encoding header, enabling detection of proxies.

Someone asked if the posted links point to legitimate sites or to spam sites. Youngsang said that they could point to either. One link went to an Amazon profile that included the link being farmed. The same person asked if this was baked into the software; Youngsang said it was up to the forum spammer. Chris Kruegel asked if most links got to spam pages or were they used for search engine optimization? Youngsang said that they didn't get a good classifications of links.

The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns

Brett Stone-Gross, University of California, Santa Barbara and LastLine, Inc.; Thorsten Holz, Ruhr-University Bochum and LastLine, Inc.; Gianluca Stringhini, University of California, Santa Barbara; Giovanni Vigna, University of California, Santa Barbara, and LastLine, Inc.

Brett Stone-Gross explained that through their continuing efforts to investigate botnets, they located over 20 command-and-control (C&C) servers for the Cutwail botnet. They had previously established relationships with various hosting providers, and after shutting down these servers, they gained access to a subset of them. These included 13 servers, three development systems (with sources), 24 databases containing billions of email addresses, spam templates, and a 40-page user manual. They also uncovered statistics on the servers that covered 50–67% of the Cutwail botnet. During 26 days in 2010, Cutwail successfully delivered (SMTP servers accepted) 87.7 billion spam emails. During the entire period covered by the captured logs, 516,852,678,718 messages were accepted for delivery out of a total of 1,708,054,952,020 attempts.

Brett described the bot life cycle as a primary infection, via some form of spam, of either Pushdo or other malware loaders. The loaders contact the C&C servers and install the bot software. Over a period of days, the bot may be removed by anti-malware tools, or become less effective at delivering spam after being blacklisted. Cutwail's operators must continually collect new bots. Within an average of 18 hours, 90% of bots have been added to blackhole lists for spamming activity. Cutwail does not assign unique IDs to bots, but one day's logs showed that there were 121,336 bots online.

The operators rent out the parts of the botnet via a Web interface that helps the purchaser design spam templates, rent email lists, rent groups of bots, and check whether SpamAssassin will accept the email as real. The costs of email lists vary based on type of address and location. Groups of bots also vary in price, depending on whether they are "clean loads" (only infected with Cutwail), country of location, and whether they have been blacklisted yet. User account information for connecting to mail servers can also be rented.

The Cutwail C&C servers also maintained an archive of Spamdot.biz, a spammer's forum. As forum participants kick out grifters and new members must be vetted by trusted members, the authors believe that the information in the forum was accurate. For example, a spam campaign's cost ranged from $100 to $500 per million emails sent. New malware installations were sold for $300–$800 for 10,000. The authors estimate that the Cutwail operators have netted between $1.7 and $4.2 million since June 2009, depending on the level of bulk discounts given to big customers.

Nathaniel Husted wondered about the costs of new bots; Brett explained that bots need constant replacing. Stefan Savage asked what the cost premium was for an exclusive load. It costs five times as much for clean loads, but how could the operators tell the load was clean? Do the operators collect a share of the proceeds of spam campaigns? They have "affiliate programs," and the operators could receive as much as 40% of the profits on these campaigns. How could the operators be sure they were getting their share? "In the underground, everything relies on trust."

On the Effects of Registrar-level Intervention

He (Lonnie) Liu and Kirill Levchenko, University of California, San Diego; Márk Félegyházi, Budapest University of Technology and Economics and International Computer Science Institute; Christian Kreibich, University of California, Berkeley, and International Computer Science Institute; Gregor Maier, International Computer Science Institute; Geoffrey M. Voelker and Stefan Savage, University of California, San Diego

Lonnie described the point of their research as an effort to understand the effectiveness of registrar-level intervention on spam. Their method was to examine WHOIS information for newly registered domains, which they collected by comparing TLD root zones and using a commercial service.

They started with CNNIC, the registrar for China. In 2007–2008, registering a domain name with CNNIC was one yuan (about $.15). In December 2009, CNNIC changed the price to 69 yuan (about $10) and initiated a policy that required real documentation (photocopy of business and Chinese registrant licenses), with a one-week warning period. Immediately, spam domain registration with CNNIC dropped, and shifted to Russian domains five weeks later.

In their second investigation they looked at eNom, a registrar infamous for being difficult to work with, and LegitScript, an organization that identifies legitimate US Internet pharmacies. LegitScript was able to get eNom to place thousands of domains into *clientHold*, a state where the domain is removed from root zone files and cannot be transferred to another registrar.

In both cases, Lonnie claimed there was a noticeable effect: an increase in the costs of spamming because domains now cost more and through the loss of domains via clientHolds. Someone pointed out that over time the price of domains will converge. He wondered if policies that support takedowns (clientHolds) would also spread. Lonnie said that it was difficult to get people to agree globally on anything. Dan Geer pointed out that the number of top-level domains (TLDs) was growing without bounds. Rik Farrow wondered if this would have much effect if the number of registrars stays the same. Stefan Savage suggested that with an increase in the number of TLDs, and thus domains, the cost of new domains would be reduced. Tyler Moore asked if they saw any further displacements after the eNom action. Lonnie said that happened near the end of their research, so they didn't see anything.

Invited Talk

Summarized by Brett Stone-Gross (bstone@cs.ucsb.edu) and Rik Farrow (rik@usenix.org)

Complex Web of Search Engine Spam

Sasi Parthasarathy, Bing

Sasi Parthasarathy described the difficulties that Microsoft faces in combating search engine spam, which poses a significant threat to the quality of search results returned by Bing. Sasi presented two primary types of search engine spam: page-level and link-level. Page-level spam relies on a number of techniques to influence the search engine rank such as keyword stuffing, parked domains, hidden/invisible text, machine generated content, and social media spam. The primary objective of these attacks is to deceive a search engine into believing that the content of the page is relevant to popular search queries, when in fact the content is low quality or potentially malicious. Link-level spam exploits the fact that Web sites are ranked higher by search engines if other sites contain links to them. As a result, attackers create sites (known as link farms) that link to their own sites and contain no useful content. In addition, link exchange communities have formed to mutually exchange links in order to boost their search rank.

Sasi also presented several newer types of threats such as hijacked sites, scareware, and low-quality user-generated content, and upcoming threats such as "like" farms. Bing currently uses both manual and algorithmic approaches to combat these search engine spam threats.

Stevens Le Blond asked about cloaking mechanisms. Sasi said that advanced cloaking has to do with recognizing the range of your IP address. Rik Farrow asked about the type of search engine spam used by a well-known retailer during the 2010 Christmas season. Sasi said that the retailer had paid a company to purchase links, moving the retailer into the number one position for many searches. Stefan Savage wondered how long someone needs to have their pages lifted in the ratings to make it worthwhile, and Sasi said while it was a great point, they don't look at those numbers. Tyler Moore asked about link farms. Sasi responded that link farms can span hundreds of sites, giving instant authority to lots of links. Moore then asked if link farms can contain legitimate links, and Sasi said that they can, and that makes them much harder to block. Dan Geer asked how long it took before a new site gets noticed. Sasi said that you won't be instantly seen, as they are looking for authority on the Web and that takes time. They also are looking for value for the user.

Nick Mathewson asked if they see any anti-optimization, attempts to get sites banned by using bad links (like links for Viagra pointing to a site). Sasi said they can't tell this from other behavior and don't punish sites for it. Chris Kruegel wondered about a site that gets lots of authority because of negative comments with links directed to a site. Does Bing use semantic analysis? Sasi said they do not police the Web, but do honor DMCA requests. Stefan asked if they follow noindex in a robots.txt file with illegal sites, and Sasi said they honor these tags. Stefan followed up by asking if they have a specific malware crawler, and Sasi said they didn't.

Threat Measurement and Modeling

Summarized by He Liu (h8liu@cs.ucsd.edu)

Characterizing Internet Worm Infection Structure

Qian Wang, Florida International University; Zesheng Chen and Chao Chen, Indiana University—Purdue University Fort Wayne

Qian Wang and his advisers attempted to characterize the underlying topology of a infection tree formed by worm infection. In particular, they are interested in modeling the children count of infected nodes, and the generation that a node belongs to. In distinction to previous work, they focused on micro-level characteristics, trying to understand individual nodes. They used a sequential growth model, applying probabilistic modeling methods. The modeling results showed that most nodes do not have a large number of children, and half of the infected nodes have no children. In addition, a typical worm tree does not have a large average path length, while the distribution of the generation approximately follows a Poisson distribution with a bell-like shape probability distribution function (PDF). They verified their model with discrete time simulations.

Their work implies that if a defender has access to some number of nodes, say A, in a P2P connected botnet, they will also have access to 3A nodes: the node itself, the parent, and one child on average. A defender can reach more bots if he or she targets nodes with the largest number of children. However, future botnets can respond with a self-stopping mechanism as a countermeasure.

In the future, they will try to use fractals as a tool to model and analyze the structure.

Nathaniel Husted raised a question about the minimum number of nodes to start analyzing using fractals, and another person also wondered if fractals are appropriate for spam tree analyzing, because an Internet spam tree may lack uniform self-similarities. Giovanni Vigna said Internet worm modeling was a topic many years ago, questioning if there is a renaissance of worm modeling. Qian said they worked on the micro level, and the work has value to Conficker C-like botnets. Giovanni asked how to find the nodes with the maximum number of children. Qian answered that they estimated a worm infection sequence, and early infected hosts might have a large number of children. Giovanni further asked about the best tradeoff for self-stopping. Qian said it might not have a huge effect on worm expansion speed.

Why Mobile-to-Mobile Malware Won't Cause a Storm

Nathaniel Husted and Steven Myers, Indiana University

Nathanial Husted explained that mobile-to-mobile malware used to spread using protocols like Bluetooth on feature phones using Symbian OS. Shifting to smartphones today, WiFi becomes available. Unlike Bluetooth, WiFi is always visible, uses transparent management traffic, and has a greater range and higher speed. To understand how smartphone malware will propagate, the presenter and his colleagues modeled and simulated human mobile users in an area in Chicago, using the SEIR (Susceptible-Exposed-Infected-Recovered) model and UdelModels. They tried both parallel and serial infection styles, and used different exposure times, susceptibility, and broadcast radius. The results showed that exposure time does not have a major effect below 120 seconds, and that susceptibility plays a much bigger role than broadcast radius. The simulation implied that current US cities still do not have the density for epidemics, and epidemics require high susceptibility rates. They concluded that mobile-to-mobile epidemics are the least of our worries.

Christopher Kruegel pointed out that they varied many parameters, but not the population density. What happens if the user density increases? Nathaniel answered that it is an interesting question and worth looking at in the future; realistic city user data with high density would be helpful. Stefan Savage asked why other vulnerabilities were not considered. Nathaniel answered that this is the vector of their focus, while other vulnerabilities may have characteristics similar to traditional PC-based malware.

Inflight Modifications of Content: Who Are the Culprits?

Chao Zhang, Polytechnic Institute of NYU; Cheng Huang, Microsoft Research; Keith W. Ross, Polytechnic Institute of NYU; David A. Maltz and Jin Li, Microsoft Research

Chao Zhang presented work in which they found that nearly 2% of clients in the US are affected by inflight modifications, and 44 local DNS servers (LDNS) in nine ISPs redirect clients to malicious servers. They collected user data from 15 million unique clients, of which 4,437 were proxies. They determine if a page is modified by comparing two pages fetched through two servers; different contents may indicate that the proxy is malicious. They discovered four different types of modifications: modifying the search result links, modifying advertisement links, inserting JavaScript, and redirecting requests. In total, they found 349 rogue servers.

Afterwards, Chao talked about the root cause of these modified pages. Some of the cases resulted from compromised local domain name servers. In fact, 48 out of 108 LDNS, grouped by /24 IP prefix, were compromised, and they were also operated by a small number of ISPs. Further study showed that these ISPs indiscriminately redirect all clients to malicious servers. Since only the DNS level is affected, public DNS improves service availability.

Christopher Kruegel asked whether the presenter thought that these ISPs are knowingly involved. Chao said that they have not contacted any ISPs to confirm that. Christian Kreibich said that he had seen ISPs that only redirect search engine Web queries, and asked whether they have similar or related observations. Tyler Moore asked if it is common to see redirections to ad-related domains. Chao said it is common, and some of the malicious servers do not redirect every time. Sasi Parthasarathy asked if there are any particular patterns of Web requests that are affected, and Chao responded that this has not been examined yet. Giovanni Vigna asked how they collected user information with user privacy as a concern. Chao answered that it is collected through MSN Toolbar from users who agreed to share data with MS to improve performance.

New Threats and Challenges

Summarized by Rik Farrow (rik@usenix.org)

Application-Level Reconnaissance: Timing Channel Attacks Against Antivirus Software

Mohammed I. Al-Saleh and Jedidiah R. Crandall, University of New Mexico

Mohammed Al-Saleh described how the authors fingerprinted AV by using timing attacks. For example, if you unzip Code Red, it takes about 100 seconds before Symantec AV recognizes it and cleans up. In their research, they used ClamAV. ClamAV filters on file writes by generating a hash and searching for this hash in the signature database. By using different files, they can use the length of time before detection to determine the update level of the AV in use.

They used ActiveX to open and close a file and to sample CPU time to determine how busy the CPU was. With this they could determine with high accuracy if a particular signature was included in an AV database. Mohammed claimed this will work against algorithmic and heuristic defenses as well.

Ben April (Trend Micro) wondered about the applications for their research. Mohammed said that they could fingerprint both the type of AV and the specific signature-update level. Ben then wondered if they had considered cloud-based AV, and Mohammed said they hadn’t considered it. Engin Kirda wondered if other AV scanners would have the same issues, and Mohammed said they focused on ClamAV as they had the source for it, but that they thought other AV scanners would have the same issues. Ben April asked about countermeasures, and Mohammed said they should do “the usual”: add in delays. With a scan taking 100 seconds already, a little more delay would hardly be noticed.

Reconstructing Hash Reversal based Proof of Work Schemes

Jeff Green, Joshua Juen, Omid Fatemieh, Ravinder Shankesi, Dong Jin, and Carl A. Gunter, University of Illinois at Urbana-Champaign

Jeff Green explained that some servers use proof-of-work protocols as resource management. These protocols require that the clients are willing to perform some amount of work. There are two types of protocols: task-based, and task plus time token–based. The typical task requires the client to perform hash reversal puzzles that are easy for the server and difficult for the client. The server provides bounds and the client must brute-force an answer.

In their research, they used an older GPU that worked with CUDA, an API for running programs on GPUs. Jeff said that even their several-year-old graphics processor was 600 times faster than last year's high-end CPU at solving these puzzles. Their suggested solution requires that servers track clients and vary the amount of time required by the puzzle based on client responses.

If they are tracking clients, Chris Kruegel wondered, wouldn't it be simpler to rate-limit clients? It would, but only if you wanted to predetermine how much service everyone gets. Chris asked if this took anything special, and Jeff answered that it only required a CUDA-enabled graphics card and an application to use it. Dan Geer remarked that work-based proofs are now a non-starter, and Jeff agreed.

Andbot: Towards Advanced Mobile Botnets

Cui Xiang, Fang Binxing, Yin Lihua, Liu Xiaoyi, and Zang Tianning, Institute of Computing Technology, Chinese Academy of Sciences

Guong Liung of Boston University presented this paper, as the authors could not attend because of visa problems. He identified himself as a friend of a friend of the authors. He attempted to explain the paper, which he suggested was better if just read.

The authors expect that smartphones will soon be used in botnets. But smartphones have specific issues, such as limits on power, generating network traffic which may reveal the bot, and the absence of a public IP address. The authors suggest several strategies to make Andbot low cost, resilient, and stealthy. For example, they use signals detected in microblogging sites (e.g., Twitter) to find JPEG images that have botnet commands embedded in them.

Sven Dietrich (Stevens Institute) wondered how this could work, as mobile providers tend to compress images and this would likely scramble any commands. He also pointed out that there are already command-and-control botnets that use Twitter. Goung again suggested reading the paper.