The Koobface Botnet and the Rise of Social Malware

Kurt Thomas
University of Illinois, Urbana-Champaign
kathoma2@illinois.edu

David M. Nicol
University of Illinois, Urbana-Champaign
dmnicol@illinois.edu

Abstract

As millions of users flock to online social networks, sites such as Facebook and Twitter are becoming increasingly attractive targets for spam, phishing, and malware. The Koobface botnet in particular has honed its efforts to exploit social network users, leveraging zombies to generate accounts, befriend victims, and to send malware propagation spam. In this paper, we explore Koobface's zombie infrastructure and analyze one month of the botnet's activity within both Facebook and Twitter. Constructing a zombie emulator, we are able to infiltrate the Koobface botnet to discover the identities of fraudulent and compromised social network accounts used to distribute malicious links to over 213,000 social network users, generating over 157,000 clicks. Despite the use of domain blacklisting services by social network operators to filter malicious links, current defenses recognize only 27% of threats and take on average 4 days to respond. During this period, 81% of vulnerable users click on Koobface spam, highlighting the ineffectiveness of blacklists.

1 Introduction

In recent years, online social networks have exploded in popularity. Today, sites such as Facebook and Twitter attract nearly 500 million members combined [7, 16], each allowing users to share photos, stories, and disseminate links. Implicit to the interactions within a social network is the notion of trust; users create relationships with their friends and valued media outlets, in turn receiving access to content generated by each relationship. On the heels of the widespread adoption of social networks, phishing and malware attacks have become a regular occurrence [8, 14], exploiting the trust users place in their friends.

Of the multitude of attacks appearing in social networks, the Koobface botnet in particular has evolved into a sophisticated infrastructure honed at exploiting social networks [4]. Leveraging its zombie arsenal, the Koobface botnet automates the creation of new social networking accounts used to befriend unsuspecting users, in turn spamming enticing links that redirect to malware. Victims that fall prey to the social engineering attack witness their own social networking accounts turn into vehicles for sending spam to the victim's friends, while the victim's machine is repurposed into a zombie.

In this paper, we explore Koobface's recent spamming activity and analyze how Koobface evades defenses implemented by social networks to prevent the spread of malware. To accomplish this task, we develop a zombie emulator that safely interacts with the Koobface C&C to acquire work loads without any risk of propagating malware. Over a month long infiltration, we discover over 1,800 compromised hosts and the identities of 4,100 zombies subverted by Koobface to serve malware. In addition to monitoring C&C activity, we identify 942 fraudulent Facebook accounts generated by Koobface and 247 infected Twitter accounts which were used to send malicious links to over 210,000 users, generating over 157,000 clicks.

Despite signs that Koobface spam is becoming less frequent, the current phase of remission is not due to protections put in place by social networks. Monitoring blacklists used by social networks to identify Koobface's malicious links, we find even the best blacklist identifies only 26% of links, requiring on average 4 days between a link being spammed to its subsequent blacklisting. During this period of delay, we find 81% of visitors to Koobface's spam occur within the first 2 days of a link being posted, leaving the majority of social networking users vulnerable. Paired with Koobface's use of URL obfuscation which can completely evade existing blacklist techniques, social networks remain largely undefended from the threat of Koobface.

2 Background

As the ingenuity of spammers continues to evolve, unsolicited messages have expanded beyond email and into social networks, posing a novel threat that remains largely unexplored. Earlier studies into botnets have targeted infiltration for improving email spam detection [15], identifying the hosting infrastructure of scams [3], understanding the economic motives of spam [10], and determining what information is stolen from infected machines [18]. While these studies form a foundation for botnet infiltration, they exclusively target systems that rely on email propagation.

Where traditional email spam relies on access to bulk lists of email addresses, social network spam requires the creation of fake user accounts or compromising existing accounts. Without access to relationships with other users, a message cannot be propagated. The challenge of a successful spam campaign in social networks is thus two fold: obtaining enough accounts to carry out a campaign before the accounts are suspended and enough URLs to evade filtering. The Koobface botnet in particular has matured to address both of these challenges.

In an attempt to stem the spread of spam, social network operators have implemented a number of safety measures that include using URL blacklisting services to identify and delete suspicious URLs, constructing heuristics to identify malicious activity and suspend the offending account, and blocking the IP addresses of repeated abusers [19, 12, 17]. Despite the array of defenses, social networks continue to be targeted by successful spam campaigns.

Given Koobface's impact on social networks, a number of researchers have previously studied the botnet, centering on its network infrastructure and the components sureptitiously installed on each zombie [2, 6, 4]. Our work expands upon this research, analyzing in depth the functionality related to Koobface's spread in both Facebook and Twitter, the ease at which the botnet recovers from takedown, and the techniques employed by the botnet to confound both security researchers and social network operators.

3 The Koobface Botnet

The Koobface botnet, which first appeared in late 2008 [11], has evolved into a complex system that preys on social networking sites as its primary means of propagation. The infection chain, described in Figure 1, begins with an unsuspecting victim browsing Facebook or Twitter being sent a message from a user they believe to be a friend. In truth, this user is either a *compromised account* that fell for one of Koobface's scams or a *fraudulent account* generated by Koobface to automatically befriended victims. Each Koobface message includes a malicious URL obfuscated by shortening services such as bit.ly or wrapped by an innocuous website including Google Reader and Blogger. Clicking on the URL initiates an elaborate chain of redirection that includes a *compromised redirector* and *zombie webhost* until a victim is finally presented with a spoofed YouTube or Facebook page that attempts to trick the victim into installing malware masquerading as a Flash update. Victims recruited in this manner then spam their own social network friends, completing the propagation cycle. To understand the individual systems that facilitate Koobface's propagation, we present an overview of Koobface's current infrastructure and zombie duties directly related to spamming.

3.1 Koobface Hierarchy

Koobface consists of a two-tiered hierarchy where each zombie connects to any one of roughly a hundred compromised hosts acting as C&C master servers that disseminate spam instructions. These exploited hosts, operated by legitimate parties and re-purposed by Koobface, simultaneously serve benign content along side Koobface C&C traffic until the host is disabled or uninfected.

Despite having the capability of operating entirely behind the master servers, Koobface maintains a fixed domain that zombies regularly contact to report uptime statistics and request links for spamming activity. The remainder of zombie requests such as downloading updates or querying for tasks are routed to the C&C masters. All communication between zombies and the C&C transpires over HTTP on port 80 with only minimal use of weak encryption.

3.2 Spamming Infrastructure

The Koobface spam chain relies on a complex system of redirection to prevent domain blacklisting by social networking sites. Working backwards from the chain presented in Figure 1, externally accessible zombies act as the final landing page for Koobface's infection chain where victims are deceived into downloading a malicious executable. Due to the unpredictable uptime of these zombies, a compromised webserver with high availability acts as a front end. Once accessed, the webserver iterates through twenty zombie IPs updated daily by the C&C in search of an operational zombie, redirecting victims to the zombie. These redirects trigger only if a browser has both Flash and JavaScript enabled, preventing lightweight crawlers from proceeding along the redirect chain.

With only a limited number of compromised webservers to act as redirectors, Koobface circumvents domain blacklisting services by obfuscating URLs before spamming them to social networks. Using content automatically generated on sites such as Blogger and Google Reader, Koobface presents social network operators with well known domains that do not appear in blacklists, but whose content contains a redirect to one of Koobface's webservers. Links to these posts can in turn be obfuscated with shortening services such as bit.ly, allowing Koobface to present social networks with thousands of constantly updated URLs which ultimately resolve to a limited number of zombies serving malware.

3.3 Zombie Duties

Due to safety measures put in place by social network operators, success of the Koobface propagation campaign hinges on obtaining fresh user accounts and ma-

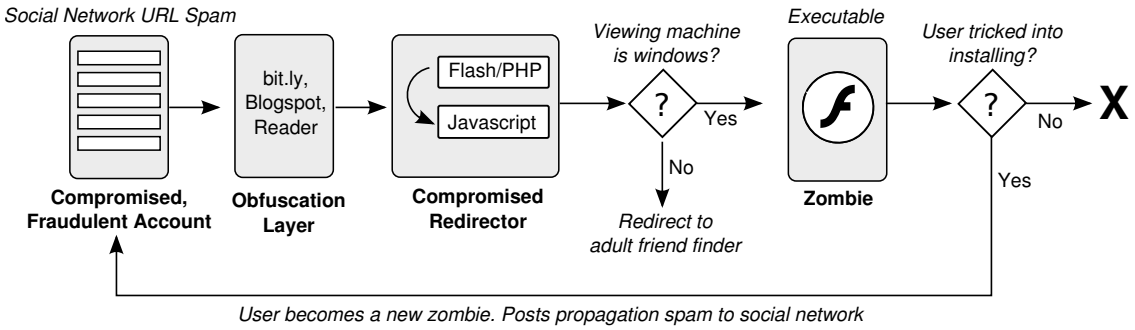

Figure 1: Koobface spamming infrastructure. Social network users are redirected through multiple layers of obfuscation until finally being presented a malicious executable to install.

licious URLs. To accomplish both tasks, zombie machines continuously poll the C&C for various duties including automated account creation, URL spamming, URL obfuscation, and Captcha solving. While Koobface operates on multiple social networking sites, we found the default zombie functionality targets Facebook for which we provide an overview.

Account Generation: One of the primary tasks of each zombie is to generate and maintain fraudulent Facebook accounts. A zombie will regularly query the C&C for login credentials to Facebook, obtaining either a command `REG`, to register a new account, or `ADD`, to login to an existing account. During registration, the C&C will provide a zombie with a randomly generated Facebook profile that includes a personal photo, birthday, background, and interests. The zombie will also be instructed to join multiple social groups based on keywords such as *Harry Potter*, *Twilight*, and other popular references to help it masquerade as a legitimate account.

For an existing account, a zombie will be tasked with acquiring new friends. To form a relationship with a user, Facebook first requires the user accept a *friend request*. The zombie will send multiple requests to random Facebook members, in turn accepting any requests made by Facebook members who have mistaken the fraudulent account as a legitimate user. Once complete, the zombie reports back to the C&C with the account's statistics. By acquiring hundreds of friends, a zombie paves the way for sending spam to victims.

URL Obfuscation: In order to obfuscate Koobface URLs, zombies are tasked with creating both Blogger and Google Reader accounts to act as redirectors. When creating a blog, a zombie will fetch the latest news headlines and generate a post containing a JavaScript redirect to a Koobface webserver. Similarly for Google Reader, a zombie creates a page containing an RSS feed provided by the C&C with an embedded redirect. The resulting links for both services are reported to the C&C, in turn obfuscated by bit.ly, and distributed to zombies for spamming. A more extensive treatment of Koobface's use of obfuscation is provided in Section 6.

Spamming Friends: To infect new hosts, zombies regularly query the C&C for malicious URLs to send to a Facebook account's friends. A Facebook account is acquired either from an infected user's machine, using the system's cookies, or provided by the C&C. Prior to spamming a URL, a zombie will first query Facebook to determine if the link is blacklisted. Non-blacklisted URLs will be spammed to all of an account's friends, while blacklisted URLs will be skipped and a new spam URL requested.

Captcha Solving: Generating Blogger, Facebook, and Reader accounts along with Gmail accounts used to register for each service requires a constant stream of solved Captchas. As described in an earlier report, Koobface pushes Captcha solving onto zombie machine users, requiring the user to input a Captcha solution under (false) threat of restarting the machine [4]. When a zombie registering for services encounters a Captcha, it sends a request to the C&C along with the image to be solved. Other zombie machines regularly poll the C&C for Captchas requiring solutions, subsequently deceiving users into solving the request and reporting the solution to the C&C.

4 Methodology

Our monitoring effort of the Koobface botnet consists of three components. The first is a manually constructed script that emulates zombie behavior, joining the Koobface botnet and polling the C&C for work. The second component targets social networking websites, logging into fraudulent accounts previously created by Koobface to monitor spamming and the efficiency of acquiring new friends. Finally, we regularly poll the Koobface C&C, compromised redirectors, and zombie webhosts to identify update cycles and uptime statistics.

4.1 Botnet Infiltration

Where previous approaches to botnet infiltration have relied on running live zombie samples in network sandboxes [10, 13, 9], we adopt an alternative approach whereby zombie behavior is reproduced by an emulator, similar to previous work in botnet detection and tracking [1, 20]. The emulator replicates communication a zombie would normally send to the Koobface C&C, while all other hostile traffic that would negatively impact the outside world remains unemulated. To construct our emulator, we acquired a number of malware executables from Koobface spam present in Facebook and Twitter, running each sample in a live virtual environment to observe Koobface's behavior. We seeded each infection with various social networking accounts and browsers, attempting to illicit a different response from Koobface for each system environment. We ran through each possible combination of:

- cookie = {facebook,twitter,none}
- browser = {ie,firefox}
- user activity = {actively browsing, dormant}

repeating each infection multiple times and storing the resulting packet traces. Zombie requests to the C&C were manually identified from the traces and subsequently replicated in our emulator, while all other traffic was ignored. The only instance of encryption in the packet traces appeared during requests to the C&C for login and password details to fraudulent Facebook accounts. To recover the decryption function, we reverse engineered the portion of a Koobface binary containing the decryption code and reimplemented the functionality in our emulator.

While construction of our Koobface emulator was tedious, the result is a functioning zombie capable of interacting with the C&C without any requirement of network sandboxing. Our fake zombie can simultaneously emulate multiple Koobface infections, replicating Twitter, Facebook, Blogger, and Gmail spam behavior which would normally require a unique infection for each tasks. Furthermore, we can run the emulator at accelerated rates compared to a typical zombie by removing all timer delays, allowing us to hone in on particularly interesting behavior.

One consequence of emulation is the need to update our system with each modification to the C&C protocol. During the course of our monitoring, we witnessed six updates to Koobface's spamming modules which added functionality to interacting with Facebook and improvements to the websever, though only one required an update to our emulator due to modifying the network protocol to include new commands. Sandboxing techniques face the same challenge of keeping pace with updates, requiring new network filters for each zombie iteration. As such, we do not feel the requirement of manual updates detracts from the benefits of zombie emulation.

4.2 Social Monitoring

To understand the impact that Koobface has on social networks, our monitoring infrastructure includes a crawler targeting Twitter and Facebook. On Twitter, we regularly search for Koobface spam strings and URLs discovered from interacting with the C&C, maintaining a list of infected accounts propagating Koobface spam. Once a Koobface Twitter account is identified, we track the account over time to measure the rate spam is sent and the average length of infection.

Due to Facebook's closed nature, the same monitoring techniques are not possible. However, using fraudulent Facebook accounts created by Koobface, we access each account and store its history of sent spam messages and the account's number of friends. The result of both approaches is a broad understanding of Koobface's social network activity from the vantage point of infected and fraudulent accounts.

4.3 Redirector Monitoring

The final component of our Koobface monitoring infrastructure targets the redirector chain of malicious URLs. Using spam URLs obtained from Koobface's C&C, we regularly poll the uptime of compromised webservers acting as redirectors and zombies hosting malware, measuring the growth and decay of Koobface's infrastructure. We extend this monitoring to include Koobface's C&C, identifying the frequency that C&C servers are shut down or move.

4.4 Dataset

Each monitoring component was executed over a month long period from January 27, 2010 through February 27, 2010. In total, we collected data from over 300 C&C servers, 4000 zombies severing as webhosts, and 1300 compromised domains acting as redirectors. In addition to the botnet's infrastructure, our data set consists of 942 fraudulent Facebook accounts provided by Koobface for spamming and 247 compromised Twitter accounts identified through crawling, each containing records of spam activity from November 2009 on through February 2010.

5 Analysis

We now present the results of our monitoring effort of the Koobface botnet, first examining properties about Koobface's infrastructure before exploring Koobface's spamming activity and the techniques it employs to generate new accounts.

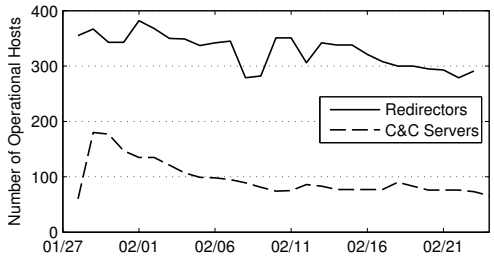

Figure 2: Number of compromised hosts per day acting as C&C servers and redirectors.

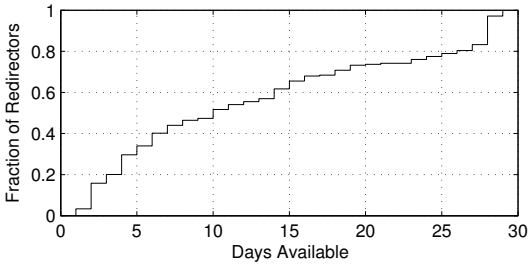

Figure 3: CDF of the lifetime of compromised hosts acting as redirectors.

5.1 Koobface Infrastructure

Koobface's reliance on compromised hosts for both C&C servers, spam redirectors, and zombies requires constant upkeep from the botnet controllers. As hosts become discovered and taken down, new hosts must be compromised to replenish lost resources. By measuring this daily churn, we find that Koobface controllers readily obtain new compromised domains to serve in the C&C and as redirectors, while a constant number of zombie webhosts remain available.

Command & Control Morphology: To discover and monitor Koobface's C&C infrastructure, we regularly emulated zombie requests to the C&C for software updates. For load balancing purposes, the Koobface C&C is a fully-connected graph where each master server is aware of every other master server. Each request to a C&C servers results in our emulator being forwarded to a second C&C to serve the request. By repeatedly querying each C&C server on a daily basis, we can walk the C&C graph, identifying new hosts and the absence of old hosts.

Over the course of our monitoring, we identified 323 compromised hosts acting as transient C&C master servers, with each server averaging a lifetime of 11 days before going silent to our update requests. Despite the decay rate, Koobface maintains an average of 97 operational servers at any time, shown in Figure 2, exhibiting an ease of obtaining new compromised hosts to participate in the C&C. During this same period, the fixed domain Koobface uses for reporting uptime statistics and acquiring account credentials never changed IPs and was consistently available.

Compromised Redirector Lifetime: Koobface's propagation campaign hinges on having highly available compromised webservers to redirect victims to malware. To discover the frequency that new domains are compromised, we polled the Koobface C&C hourly with our emulator to discover new redirector URLs that would otherwise be posted in spam. In total we identified 1802 redirector URLs served on 1390 distinct domains. On average, we discovered 20 new redirectors each day, with the total number available on any day shown in Figure 2.

To understand the susceptibility of redirectors to discovery and take down, we monitored the delay between the C&C advertising a new URL to the time the page is removed, shown in Figure 3. We found that fewer than 50% of compromised redirectors are operational for 11 days. During this period of availability, compromised hosts were re-seeded each day with a new set of zombies to forward visitors, allowing each redirector to maintain an up to date list of newly infected zombies while removing machines that have become uninfected.

Zombie Lifetime: To understand the volume of zombies serving Koobface malware, we extract the list of zombie IPs contained in the HTML served by each compromised redirector on an hourly basis. Over the course of monitoring, we identified 4,151 unique IP addresses from 80 countries used to serve malware. This does not represent the overall size of the botnet, but rather the number of zombies converted into webhosts with potentially dynamic IPs.

After identifying the IP address of a zombie, we attempt to download the malicious executable being served at hour intervals to determine whether the zombie is online. If at any point during the day a zombie serves malware, we consider it to be operational. Despite identifying hundreds of new IPs each day, as shown in Figure 4, on average only 365 zombies responded to our download requests each day, indicating new IPs may be added even if they are inaccessible from an external network, or the IPs reference dynamically located zombies that have since switched IPs and become stale. Compared to 60,000 zombie webhosts previously reported by TrendMicro [4], our results show a severe reduction in the number of zombies serving Koobface malware, indicating either a period of severe decline or a reduction in the number of zombies converted into functional webhosts.

5.2 Spamming Activity

To understand the effectiveness of Koobface's propagation throughout social networks, we monitored its activ-

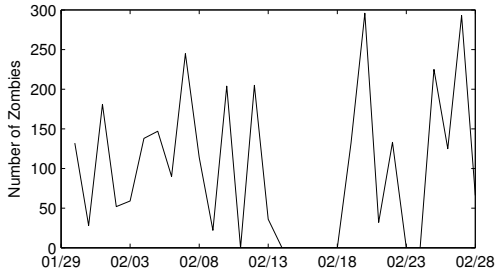

Figure 4: Arrival rate of new zombie IPs.

Spam Statistics	Facebook	Twitter
Accounts in dataset	942	259
Total friends	200,515	13,001
Total messages	506	2,847
Unique messages	476	13
Total clicks	157,399	-

Table 1: Statistics for accounts participating in Koobface's spam propagation.

ity throughout Facebook and Twitter. Using spam histories recovered from both sites, we are able to reconstruct an image of Koobface's activities from November on through February, showing reduced activity by the botnet towards later months.

Facebook: To discover Facebook accounts used for spamming, we regularly queried the Koobface C&C for account credentials. From our monitoring of the Koobface botnet, we identified that Koobface maintains a queue of Gmail accounts that is actively fed by zombies registering new accounts. This queue is subsequently accessed by other zombies tasked with either registering new Facebook accounts using Gmail addresses or for logging in to existing accounts for maintenance and spamming. By emulating the commands sent by Facebook workers, we recovered over 30,000 operational Gmail accounts, of which only 990 had yet been tied to Facebook accounts by zombies.

Before logging into these Facebook accounts, we first verify each account is fraudulent rather than stolen to mitigate any privacy or ethical issues. To the best of our knowledge, Koobface does not steal passwords; it relies on browser cookies being present from a social networking site in order to hijack a real users account. Nevertheless, we analyze each login to determine whether it matches patterns present in accounts generated by Koobface zombies. Every Koobface-generated login at the time of our monitoring follows one of two templates. The first consists of 7-15 random lower case alphabetic characters, while the second consists of two to three names separated by periods followed by two digits. Given the difficulty in distinguishing potentially legitimate accounts from the second template, we disregard

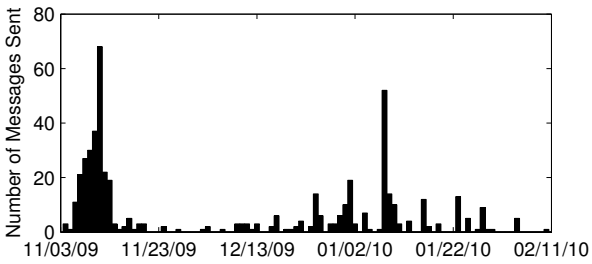

Figure 5: Number of spam messages sent by Facebook accounts per day.

48 logins out of caution. The remaining 942 Facebook logins follow the first template and are assumed to be fraudulent which we confirm upon login.

To recover spam perpetrated by Koobface, we access each Facebook account to save its list of friends and all previously sent messages. We manually analyze each outbox to verify *all* outbound messages are spam, confirming the accounts were never used legitimately. The statistics tied to these accounts can be seen in Table 1. Each fraudulent account was able to deceive an average of 202 users into following the bot, accumulating over 200,515 friends in total. Given that distributing a link to friends does not imply it will be clicked, we are able to analyze clickthrough data associated with 73% of distinct spammed links due to their obfuscation with bit.ly. Using bit.ly's statistical API, we found Koobface's spam links were clicked 137,698 times, with each link averaging 474 clicks. Despite the low volume of spam sent, Koobface accounts are still able to entice thousands of visitors.

Of particular interest is whether the Koobface botnet is increasing or decreasing in its activity. By using the timestamps associated with each spam message sent, we reconstruct a timeline of Koobface activity from November 2009 on through February 2010 shown in Figure 5. The majority of spam sent appears in November of 2009, with the frequency tapering off in later months until a brief reprisal in January. This trend of decreased activity after November is also mirrored in our Twitter data.

Twitter: While the majority of Koobface's resources are spent on Facebook, infected machines with existing Twitter cookies are re-purposed into Twitter spammers. Using Koobface URLs and messages returned by the C&C to our Twitter zombie emulator, we perform regular searches for these values using Twitter's API to identify accounts propagating Koobface spam. These accounts can be verified as infected due to non-Koobface messages appearing prior and during infection. Our search effort uncovered 247 infected accounts, the details of which are summarized earlier in Table 1. Compared to Facebook, Koobface makes no effort to obfuscate URLs spammed on Twitter or vary the messages

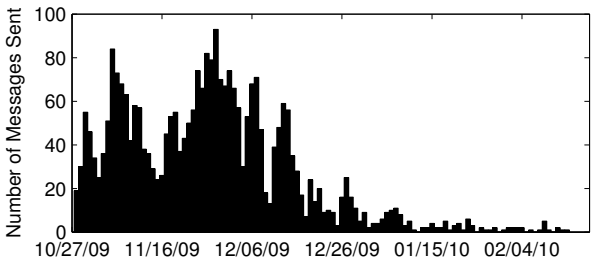

Figure 6: Number of spam messages sent by Twitter accounts per day.

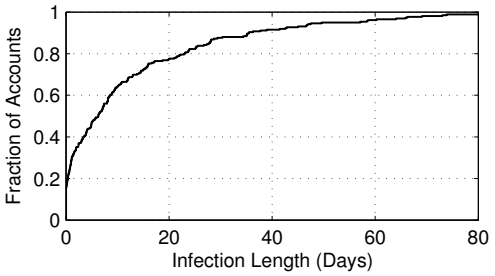

Figure 7: Length of Koobface infections for Twitter accounts.

posted to avoid spam filtering. Despite this fact, 2,847 messages were successfully spammed by infected Twitter accounts to 13,001 friends. As URLs were not obfuscated, actual clickthrough statistics are not available from bit.ly.

Collecting the history of messages posted by infected accounts, we are able to reconstruct a view of Koobface's spamming activity in Twitter, presented in Figure 6. November and December saw the brute of Koobface's activity, followed by steep drop throughout January and February. To understand the root of this cause, we examined the average length a Twitter account is abused, comparing the elapsed time between the first and last message spammed. Figure 7 shows that while 10% of infections last over a month, the majority of infections last under 6 days. The drop in Twitter activity can thus be interpreted as a failure of Koobface to acquire new infections as older zombies become uninfected. If true, this would also explain the decline in Facebook activity as fewer zombies are available for spamming tasks.

6 Evading Detection

The primary defense leveled by social networking websites against Koobface's malware propagation is the use of domain blacklisting services. In this section, we explore the limitations of blacklists due to Koobface's use of URL obfuscation and the general delay between a link being spammed and it subsequent blacklisting.

Technique	Sample
None	http://www.compromised.ca/{*path*}/
bit.ly	http://bit.ly/{*id*}
bit.ly	http://{*ip: binary,int,hex,octet*}/{*id*}
Reader	http://google.{*tld*}/reader/shared/{*id*}
Blogger	http://{*screen name*}.blogspot.com/

Table 2: Obfuscation techniques employed by Koobface.

Blacklist	Number Detected	Detection Rate
Google	144	26.71%
SURBL	31	5.70%
Joewein	0	0.00%

Table 3: Blacklist detection rate for URLs spammed by Koobface

6.1 Obfuscation Techniques

To prevent the spread of malicious URLs, both Twitter and Facebook rely on blacklists to identify and block suspicious domains; Twitter uses Google's Safebrowsing API [12], while Facebook relies on its own proprietary blacklist [17]. To evade blacklist detection, Koobface will employ any one of multiple obfuscation techniques, presented in Table 2. By using blogs, RSS feeds, and shortened URLs that forward users to compromised redirectors, Koobface masks domains known to host malware with sites that have yet to be blacklisted. Over the course of one week monitoring Koobface's obfuscation activity, our emulator recovered 3,052 bit.ly URLs spammed by Koobface that resolved to only 113 compromised redirectors. In addition, our emulator recovered 30,193 Gmail accounts used for generating malicious blogs and RSS feeds. To confirm Koobface's obfuscation techniques negate blacklists, we gathered 500 URLs blacklisted by both Twitter and Facebook and shortened each with bit.ly before resubmitting each link to check on its blacklist status. For both sites, all 500 links went unflagged as malicious, requiring both sites to eventually update their blacklists to detect the malicious URLs. While bit.ly disables links to malicious pages using its own blacklists derived from Google and SURBL [5], using bit.ly negates the blacklists employed by Twitter and Facebook. Unless Facebook and Twitter update their services to resolve URL redirects to identify a link's final landing page, obfuscation will continue to pose a threat to social network defenses.

6.2 Blacklist Delay

The reliance of social networks on blacklist for identifying malicious content requires blacklists to quickly update in response to threats. To measure blacklist delay,

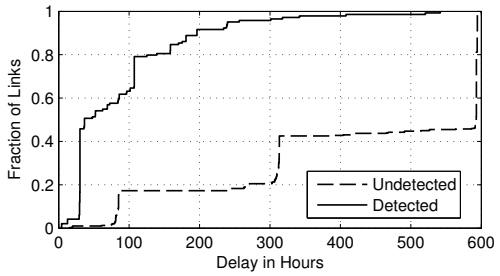

Figure 8: CDF of the delay between a URL being spammed and its subsequent blacklisting.

we monitored the time between a new URL being advertised for spamming and its subsequent appearance in three blacklist services: Google Safebrowsing, SURBL, and Joewein. Using a dataset of 544 previously unspammed compromised redirectors that were not blacklisted when our test began, we monitored each domain's blacklist status from the time it was first distributed by the Koobface C&C for spamming. The overall detection rate from our test can be seen in Table 3. The failure of SURBL and Joewein to identify Koobface's malicious URLs is likely a result of their use of email to seed blacklists, while Koobface exclusively targets social networks. Google performs the best of all blacklists, but over 73% of all malicious links went undetected.

The delay in detection for Google Safebrowsing can be seen in Figure 8, which shows 50% of links were detected in under two days. Conversely, 50% of links that have yet to be detected have been in our system over 25 days. To understand how quickly blacklists must respond, we examined the clickthrough statistics provided by bit.ly for URLs spammed in Facebook. Due to requiring manual analysis, we selected a random sample of 75 URLs from the 290 URLs spammed by fraudulent Facebook accounts. Clickthrough rates exhibited a power law distribution, with 55% of clicks appearing on average within the first day and 81% of clicks within the first two days, before tapering out into a long tail. Assuming the distribution of clicks remains constant for each Koobface URL spammed, blacklists must respond to threats within 2 days to protect the majority of users. Of the 144 URLs blacklisted by Google, only 74 blacklistings occurred within 48 hours, 13% of all URLs spammed by Koobface. Even in the absence of obfuscation techniques used by Koobface, simply using Google's Safebrowsing API, SURBL, or Joewein is ineffective in stemming the spread of Koobface.

7 Conclusion

As millions of users continue to flock to online social networks, sites such as Facebook and Twitter are becoming increasingly attractive targets for spam, phishing, and malware. The Koobface botnet in particular has honed its efforts to exploit social network users, leveraging zombies to generate accounts, befriend victims, and to send spam. Despite defenses put in place by social network operators, domain blacklisting remains ineffective at quickly identifying malicious URLs, taking on average 4 days to respond to threats, while 81% of users visit Koobface URLs within 2 days. To stem the threat of Koobface and the rise of social malware, social networks must advance their defenses beyond blacklists and actively search for Koobface content, potentially using infiltration as a means of early detection.

References

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, page 52, 2006.

[2] abuse.ch. Koobface the social network trojan. 2009. `http://www.abuse.ch/?p=2103`.

[3] D. Anderson, C. Fleizach, S. Savage, and G. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–14. USENIX Association, 2007.

[4] J. Baltazar, J. Costoya, and R. Flores. The Heart of KOOBFACE C&C and Social Network Propagation. 2009.

[5] bit.ly. Spam and Malware Protection. 2009. `http://blog.bit.ly/post/138381844/spam-and-malware-protection`.

[6] Dancho Danchev. Dissecting Koobface Worm's Twitter Campaign. 2009. `http://ddanchev.blogspot.com/2009/07/dissecting-koobface-worms-twitter.html`.

[7] Facebook. Statistics, 2009. `http://www.facebook.com/press/info.php?statistics`.

[8] D. Ionescu. Twitter Warns of New Phishing Scam. *PCWorld*, 2009.

[9] J. John, A. Moshchuk, S. Gribble, and A. Krishnamurthy. Studying spamming botnets using Botlab. In *Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[10] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, 2008.

[11] G. Keizer. Worm spreads on Facebook, hijacks users' clicks. *Computerworld*, 2008.

[12] Kim Zetter. Trick or Tweet? Malware Abundant in Twitter URLs. *Wired*, 2009.

[13] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. On the spam campaign trail. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET08)*, 2008.

[14] E. Mills. Facebook hit by phishing attacks for a second day. *CNET News*, 2009.

[15] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. Voelker, V. Paxson, N. Weaver, and S. Savage. Botnet Judo: Fighting Spam with Itself. In *Proc. of the 17th Annual Network and Distributed System Security Symposium (NDSS)*, 2010.

[16] E. Schonfeld. Twitter Reaches 44.5 Million People Worldwide In June (comScore). *TechCrunch*, 2009.

[17] B. Stone. Facebook Joins With McAfee to Clean Spam From Site. *New York Times*, 2010.

[18] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: Analysis of a botnet takeover. *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009.

[19] Twitter. The Twitter Rules. 2009. `http://help.twitter.com/forums/26257/entries/18311`.

[20] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda. Automatically generating models for botnet detection. *ESORICS*, pages 232–249, 2010.