

Συστήματα και Τεχνολογίες του Διαδικτύου HY558

Άνοιξη 2010

Θανάσης Πέτσας, (petsas@csd.uoc.gr)

AM: MET. 600

Your Botnet is My Botnet: Analysis of a Botnet Takeover

Σε αυτό το paper οι συγγραφείς παρουσιάζουν την προσπάθειά τους στο να πάρουν υπό την κατοχή τους το Torpig Botnet και να μελετήσουν τη λειτουργικότητά του για μια περίοδο 10 ημερών.

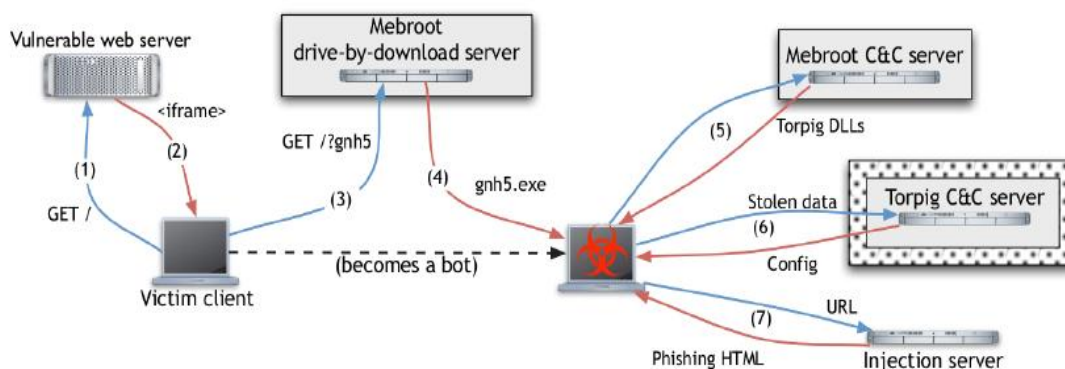
Στην αρχή οι συγγραφείς αναφέρουν τις δυσκολίες που μπορεί να συναντήσει κανείς όταν μελετάει ένα botnet. Ένα botnet είναι ένα δίκτυο που αποτελείται από μολυσμένα μηχανήματα (pcs), τα οποία είναι υπό τον έλεγχο ενός, που συνήθως αποκαλούμε botmaster, και μπορούν να δέχονται εντολές από αυτόν και να εκπληρώνουν τους κακόβουλους σκοπούς του. Bot ονομάζεται ένας τύπος malware που σκοπό έχει να μολύνει όσο το δυνατόν περισσότερα Internet hosts. Όταν μολυνθεί ένα host με ένα bot τότε κάνει join το botnet. Γενικά υπάρχουν διάφοροι τρόποι να μελετήσει κανείς ένα botnet. Ένας τρόπος είναι με *passive analysis*. Σύμφωνα με αυτή τη προσέγγιση μπορεί κάποιος να παρακολουθήσει τη δραστηριότητα ενός μολυσμένου μηχανήματος για να μελετήσει τα διάφορα secondary effects. Για παράδειγμα μπορεί να εξετάσει τα spam mails που πιθανόν να προέρχονται από bots, να εξετάσει DNS queries ή DNS blacklist queries, ή να αναλύσει network traffic σε επίπεδο tier-1 ISP για να βρει συμπεριφορές που ανήκουν σε συγκεκριμένα botnets. Το μειονέκτημα εδώ είναι ότι το detection είναι περιορισμένο σε botnets που παρουσιάζουν τη συγκεκριμένη δραστηριότητα που ψάχνουμε κάθε φορά. Μια άλλη active προσέγγιση είναι μέσω infiltration, όπου κάποιος μπορεί να χρησιμοποιήσει το ίδιο το malware, ή έναν client που προσομοιώνει ένα bot για να κάνει join ένα botnet και να δει τι είδους traffic ανταλλάσσεται ανάμεσα σε αυτό και τον C&C server (command and control).

Οι attackers δυστυχώς έχουν προσαρμοστεί σε αυτές τις μεθόδους και μπορούν να τις αντιμετωπίσουν πλέον χρησιμοποιώντας stripped-down IRC ή HTTP servers για τα command & control channels τους. Για να ξεπεράσει κάποιος αυτούς τους περιορισμούς και να καταφέρει να κάνει μετρήσεις σε ένα botnet μπορεί να κάνει "hijack" (να αρπάξει δηλαδή υπό την κατοχή του) τα μηχανήματα που κάνουν host τους command & control servers. Τα bots κάνουν συνήθως resolve ένα domain στο οποίο βρίσκεται ο C&C server με τον οποίο θα επικοινωνήσουν. Οπότε για να καταφέρει κάποιος να κάνει "hijack" το C&C server ενός botnet μπορεί να έρθει σε συνεννόηση με domain registrars και να αλλάξει το mapping του botnet domain έτσι ώστε να δείχνει σε ένα δικό του μηχανήμα. Παλιότερα τα botnets χρησιμοποιούσαν IP fast-flux τεχνικές, όπου τα bots έκαναν resolve ένα domain το οποίο έκανε map σε ένα σύνολο από IP addresses για περισσότερη ευελιξία από την μεριά του attacker (αν κάποια IP address π.χ. γινόταν blacklisted). Αυτό όμως προϋποθέτει single point of failure αφού χρησιμοποιείται ένα μόνο domain name. Σήμερα κάποια botnets

όπως το Torpig λύνουν αυτό το πρόβλημα χρησιμοποιώντας **domain flux**. Με το domain flux τα bots κατασκευάζουν περιοδικά και ανεξάρτητα μία λίστα από domain names τα οποία μετά ελέγχουν ένα-ένα (προσπαθούν να συνδεθούν δηλαδή με αυτά) για το αν βρίσκεται εκεί ο C&C server περιμένοντας να λάβουν ένα valid response. Κάνοντας κανείς reverse engineering στο malware executable μπορεί να κατανοήσει (ανακατασκευάσει) αυτόν τον αλγόριθμο DGA (Domain Generation Algorithm) και να προσπαθήσει να κάνει register πρώτος ένα domain με το οποίο θα επικοινωνήσουν τα bots στο μέλλον, πριν προλάβει να το κάνει αυτό ο attacker (botmaster). Αυτό κατάφεραν και οι συγγραφείς για το Torpig botnet. Στη συνέχεια θα αναφέρουμε πως δουλεύει το Torpig botnet, ποιος είναι ο DGA που χρησιμοποιεί, καθώς και την ανάλυση που έκαναν οι συγγραφείς στις 10 μέρες που είχαν υπό την κατοχή τους το botnet.

Το Torpig κλέβει ευαίσθητη πληροφορία από τα θύματά του και την στέλνει πίσω σε αυτούς που το ελέγχουν.

Διαδίδεται στα θύματά του σαν μέρος του Mebroot, το οποίο είναι ένα rootkit, που παίρνει τον έλεγχο του μηχανήματος αντιγράφοντας τον εαυτό του στο MBR(Master's Boot Record) του συστήματος. Αυτό του επιτρέπει να εκτελείται κατά το boot time, πριν την φόρτωση του λειτουργικού συστήματος και καθιστά δύσκολο τον εντοπισμό του από τα περισσότερα anti-virus tools. Στη παρακάτω εικόνα φαίνεται ο κύκλος ζωής του torpig Botnet:



Τα θύματα μολύνονται με **drive-by-download attacks**. Αυτές οι επιθέσεις γίνονται χωρίς την γνώση του χρήστη. Σε αυτές τις επιθέσεις νόμιμες σελίδες που παρουσιάζουν αδυναμίες (vulnerabilities), τροποποιούνται (τους προσθέτονται κάποια HTML tags) (1) έτσι ώστε να αναγκάζουν το θύμα να κάνει request κώδικα JavaScript (2) από ένα άλλο web site (*drive-by-download server*) που ανήκει στον attacker (3). Αυτός ο JavaScript κώδικας εφαρμόζει μια σειρά από exploits στον browser του θύματος. Αν πετύχει κάποιο από αυτά τα exploits, τότε ένα executable γίνεται download από το drive-by-download server στο θύμα και εκτελείται (4). Αυτό το executable είναι ουσιαστικά ο installer του Mebroot, ο οποίος κάνει inject ένα DLL στο file manager process (explorer.exe) και η εκτέλεση συνεχίζεται στο context του file manager, δηλαδή όλα τα actions είναι σαν να γίνονται από legitimate κώδικα. Ο installer κάνει include ένα kernel driver που κάνει wrap το original disk driver (driver.sys). Σε αυτό το σημείο ο installer κάνει load ένα kernel driver και κάνει overwrite το MBR του μηχανήματος με το Mebroot και μετά από λίγα λεπτά κάνει reboot το μηχάνημα. Μετά το reboot, το Mebroot επικοινωνεί με το Mebroot C&C server για να πάρει κάποια

malicious modules (5). Κάθε 2 ώρες το Mebroot συνδέεται με το C&C server για πιθανά updates. Όλη η επικοινωνία γίνεται μέσω HTTP με sophisticated encryption. Κατά το monitoring οι συγγραφείς παρατήρησαν ότι ο C&C server έστειλε 3 modules που αποτελούν το Torpig malware. Το Mebroot κάνει αυτά τα modules (DLLs) σε έναν αριθμό απλό εφαρμογές, όπως web browsers (e.g., Explorer, Firefox, Opera), FTP clients (e.g., CuteFTP, LeechFTP), email clients (e.g., ThunderBird, Outlook, Eudora), Instant messengers (e.g., Skype, ICQ), και προγράμματα συστήματος όπως το cmd.exe. Μετά το injection το Torpig μπορεί να παρακολουθεί τα data αυτών των εφαρμογών για να εντοπίσει ενδιαφέρουσα πληροφορία (online accounts, stored passwords κτλ.). Κάθε 12 λεπτά επικοινωνεί με το C&C για να κάνει upload τα data που έκλεψε από την τελευταία φορά (6). Επίσης, το Torpig πραγματοποιεί “man-in-the-browser” phishing attacks για να αποσπά επιπλέον πληροφορία από τους χρήστες που δεν είναι δυνατόν να συλλέξει μέσω του passive monitoring που κάνει. Ο C&C server στο configuration file που στέλνει ανά τακτά χρονικά διαστήματα, καθορίζει και μια λίστα από domains που θα χρησιμοποιηθεί από το torpig για τα phishing attacks. Αρχικά όταν ένα μολυσμένο μηχάνημα επισκέπτεται ένα domain που υπάρχει στο configuration file (π.χ. ένα site τράπεζας), τότε το Torpig κάνει ένα request στον injection server. Ο server απαντάει με μία σελίδα στο συγκεκριμένο domain στην οποία θα ξεκινήσει η επίθεση (*trigger page*, συνήθως η login σελίδα ενός site), και ένα URL που περιέχει το phishing περιεχόμενο. Μετά, όταν ο χρήστης επισκεφθεί την trigger page, τότε το Torpig ζητάει το injection URL από τον injection server κάνει inject το επιστρεφόμενο content στον browser του χρήστη (7).

Ο DGA αλγόριθμος που χρησιμοποιεί το Torpig δέχεται σαν είσοδο την τρέχουσα ημερομηνία και μια αριθμητική παράμετρο. Ο αλγόριθμος πρώτα υπολογίζει ένα “weekly” domain name *dw* που εξαρτάται από την τρέχουσα εβδομάδα και τον τρέχον χρόνο, και είναι ανεξάρτητο από την τρέχουσα μέρα. Μετά το bot κάνει append σε αυτό το domain έναν αριθμό από TLDs (top level domains) *dw.com*, *dw.net* και *dw.biz*. Μετά κάνει resolve ένα-ένα αυτά τα domains και προσπαθεί να συνδεθεί στο C&C server του. Αν αποτύχουν όλες οι προσπάθειες, τότε υπολογίζει ένα καινούργιο “daily” domain name αυτή τη φορά *dd*, το οποίο εξαρτάται και από την τρέχουσα ημέρα. Κάνει και σε αυτό append τα TLDs που αναφέραμε και ακολουθεί επίσης την ίδια διαδικασία. Αν αποτύχουν και αυτές οι προσπάθειες τότε δοκιμάζει να επικοινωνήσει με domains που είναι hardcoded στο configuration file.

Τα bots επικοινωνούν με το C&C server μέσω HTTP POST request. Τα requests που στέλνουν έχουν ένα URL που αποτελείται από το bot identifier και το submission header με πεδία: timestamp που υποδηλώνει πότε το configuration file έγινε update την τελευταία φορά (*ts*), την IP address του bot (*ip*), τα port numbers που χρησιμοποιεί για HTTP και SOCKS proxy (*hport*, *sport*) την έκδοση του λειτουργικού συστήματος και locale πληροφορία (*os*, *cn*), το bot identifier (*nid*) και το build και version number του Torpig (*bld*, *ver*).

Οι συγγραφείς κατά τις 10 μέρες που είχαν υπό τον έλεγχό τους το Torpig, συγκέντρωσαν 70 GB από δεδομένα.

Οι μετρήσεις που έκαναν οι συγγραφείς αφορούσαν το μέγεθος του botnet (*footprint* και *live population*). Είναι από τις πιο αξιόπιστες μετρήσεις που έχουν γίνει αφού οι συγγραφείς

μπορούσαν να βλέπουν πόσα ακριβώς bots συνδέονταν στον C&C server τους και να καθορίσουν μοναδικά τον αριθμό τους από τα requests που έκαναν τα bots (nid και διάφορα headers). Παρατήρησαν οι ήταν ότι IP addresses ήταν περισσότερες από τα bots και αυτό το οφείλουν στο γεγονός ότι οι ISPs ανακυκλώνουν τις IP διευθύνσεις των πελατών τους (μέσω DHCP), οπότε ένα bot μπορεί να αντιστοιχεί σε περισσότερες από μία IP διευθύνσεις. Επίσης, υπολόγισαν τον αριθμό των κλεμμένων λογαριασμών από financial site

κυρίως και κατέγραψαν τα νέα infections που έγιναν κατά τη διάρκεια αυτού του διαστήματος. Τέλος εξέτασαν τους διαφορετικούς τύπους δεδομένων που έκλεβαν τα bots (Mailbox accounts, Email, Form data, HTTP, FTP, POP και SMTP accounts καθώς και Windows passwords). Στους κωδικούς που συγκέντρωσαν παρατήρησαν ότι περίπου το 30 % των χρηστών ξαναχρησιμοποιούσε τα ίδια credentials για να κάνει access διαφορετικά web sites και επιπλέον μεγάλο ποσοστό των χρηστών χρησιμοποιεί weak κωδικούς που μπορούν εύκολα να γίνουν break από διάσημα password cracker tools.