Department of Computer Sciences
University of Salzburg

Seminar aus Informatik

SS 2013

Darknet / Dark Cloud

19. Juli 2013

Autor: Oliver Janner[1], Jakob Reissig[2]

Betreuer: Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang Pree[3]

Universität Salzburg
Fachbereich Computerwissenschaften
Jakob–Haringer–Straße 2
A–5020 Salzburg

[1] ojanner@cosy.sbg.ac.at - Matr.Nr.: 1020187
[2] jreissig@cosy.sbg.ac.at - Matr.Nr.: 0920457
[3] office@cs.uni-salzburg.at

Contents

1 Allgemein

Was ist Cloud?

Der Begriff Cloud stammt aus dem Marketingbereich. Er umfasst viele Technische Aspekte, in denen mehrere zusammengeschlossene Rechner ein Netzwerk bilden, welches folglich Cloud genannt wird. Es ist nun möglich dieses Netzwerk auf unterschiedlichsten Ebenen zu verwenden. Direkt auf Basis des Betriebssystems oder auf diversen Applikationen von vorhandenen Systemen aufsetzend.

Was ist ein Darknet?

Ein Darknet ist ein Netz bzw. ein System. In den Medien wird das Darknet meist als eine Möglichkeit propagiert illegale Aktivitäten im Schatten des Internet abwickeln zu können. Dieser Aspekt wird jedoch in keinem wissenschaftlichen Artikel angeführt. Es wird allgemein nur von einem Netz bzw. System gesprochen, sowie den Nutzern und Verwendungsmöglichkeiten - es wird jedoch keine Wertung über die Inhalte der Systeme vorgenommen.

2 Cybercrime

Am Anfang herrschte die Netzfreiheit, dicht gefolgt von einzelnen Rechnerverbunden wie dem ARPANET, welches ein wissenschaftliches Netzwerk war. Kurz darauf konnte man mittels dem Telefonnetz eine globale Vernetzung erreichen, welche zum heutigen Zeitpunkt als das Internet bekannt ist. Hier interagieren einzelne Server, um die Anfragen der Benutzer dieses Netzes abzuarbeiten und die geforderten Informationen zu liefern.

Angriffstechnisch gab es schon in den frühen Jahren der Netzwerke viele Möglichkeiten - sei es nur der Drang Möglichkeiten auszuloten und Aktionen zu testen oder bereits der Versuch sich finanzielle Vorteile zu verschaffen. Als Beispiel kann man schon den damals bekannten Hack nennen, bei dem Töne in einer gewissen Reihenfolge und Frequenz über das Telefonnetz gesendet wurden. Als Eingabemedium diente ein einfacher Telefonhörer und als Eingabegerät wurde eine Pfeife verwendet. Dadurch erreichte man, dass Ferntelefonate ohne Münzeinwurf freigeschalten wurden oder unbegrenzt lange Telefonate möglich waren.

Im Laufe der Zeit wurden solche Lücken geschlossen, jedoch auch neue Systeme an das Netz gehängt. Diese Systeme waren nun Ziel diverser Angriffe, da dort meist finanziell lukrative Daten abgegriffen werden konnten.

Angreifer nutzen meist Spam, um sich durch die vom Benutzer daraufhin ausgeführten Aktionen einen Vorteil zu verschaffen. Jedoch sind auch direkte Angriffe auf den Rechner des Benutzers oder - wenn kein spezifischer Benutzer das Ziel ist - auch auf einen Webserver möglich.

Es gibt mehrere Möglichkeiten den Rechner eines Benutzers mit Hilfe einer kom-

promitierten Webseite zu infizieren. Zuerst gibt des den Drive-by-Download, bei dem:

- eine Lücke im Browser des Benutzers,
- oder eine Lücke in einem externen Plugin wie Flash oder Java

ausgenutzt wird, um die Schadsoftware auf den Rechner des Benutzers zu kopieren.

Schlägt die eben erwähnte Methode fehl so kann man den Benutzer auffordern die Malware selbstständig zu installieren. Etwa durch die Anzeige eines falschen Videoplayers, welcher dazu auffordert einen speziellen Codec zu installieren, um das gewünschte Video zu betrachten. Eine andere weit verbreitete Methode ist die Anzeige eines falschen Antivirus-Scanners, welcher einige Funde anzeigt. Da kein direkter Zugriff auf die Festplatte möglich ist, ist das die Unwahrheit - jedoch wissen dies die meisten Benutzer nicht und erfreuen sich an den Funden, da das angezeigte Programm auch prompt eine Lösung liefert: Das Herunterladen einer Software. Wie auch zuvor - nur direkt durch den Benutzer - wird die Schadsoftware von der Webseite heruntergeladen und installiert. Sollten die Entwickler der Software vermeiden wollen, dass die Benutzer ihren Fehler erkennen, so ist es teilweise möglich, dass nach der Installation erwartete Änderungen eintreten. So ist z.B. manchmal der heruntergeladene Virenscanner tatsächlich ein Virenscanner, der jedoch mit einigen Backdoors versehen ist.

Als Motivation für Angreifer kann man folgende Punkte aufzählen:

- Keylogger
- Backdoor
- Botnetz
- Sammeln von Daten
- Senden von Spam
- Profit

Meist versuchen die Angreifer ihren Zugriff möglichst geheim zu halten. D.h. dass der Administrator sowie die Besucher der Seite die Änderungen nicht bemerken. Somit bleibt die Änderung möglichst lange aktiv und die Anzahl der infizierten Benutzer steigt an.

Angriffe auf Server werden häufig durch einen Angriff auf eine Lücke in der Kommunikation des Webservers mit der SQL-Instanz durchgeführt. Diese und ähnliche Angriffspunkte sind durch die weite Verbreitung von OpenSource-Software sehr einfach auszunutzen.

Angriffe zielen nicht direkt darauf ab Logindaten zu erhalten, sondern eher darauf den Login zu umgehen und sich unauthorisiert als Benutzer bzw. Administrator auf der Seite anzumelden und so Inhalte auf der Seite bzw. in der Datenbank zu ändern.

Hier ein Beispiel einer SQL-Injection, welche z.B. durch ein ungeschütztes Loginfeld vom Datenbankserver ausgeführt werden kann:

```
DECLARE @T VARCHAR(255),@C VARCHAR(255)
DECLARE Table _ Cursor CURSOR FOR SELECT a.name,b.name
FROM sysobjects a,syscolumns b
WHERE a.id=b.id AND a.xtype='u'
AND (b.xtype=99 OR b.xtype=35
OR b.xtype=231 OR b.xtype=167)
OPEN Table _ Cursor FETCH NEXT FROM Table _ Cursor INTO @T,@C
WHILE(@@FETCH _ STATUS=0)
BEGIN EXEC('UPDATE ['+@T+']
SET ['+@C+']=RTRIM(CONVERT(VARCHAR(4000),['+@C+']))+'''')
FETCH NEXT FROM Table _ Cursor INTO @T,@C
END CLOSE Table _ Cursor
DEALLOCATE Table _ Cursor
```

Das oben gezeigte Beispiel fügt in jeder Tabelle, in der eine Spalte ein gewisses Schema erfüllt, ein Codefragment ein (hier nur ""). Diese Injection ist auf einen SQL-Server abgestimmt, jedoch existieren ähnliche Methoden auch in anderen Datenbanksystemen.

Um nicht die Seite selbst zu ändern, kann eine Weiterleitung eingerichtet werden, welche nicht leicht zu erkennen und genau so schwer zu verfolgen ist. Man kann diese Veränderungen sogar relativ einfach vor dem Besitzer oder Administrator verbergen:

```
RewriteEngine On
RewriteCond %{HTTP _ REFERER} .*google.*$ [NC,OR]
RewriteCond %{HTTP _ REFERER} .*aol.*$ [NC,OR]
RewriteCond %{HTTP _ REFERER} .*msn.*$ [NC,OR]
RewriteCond %{HTTP _ REFERER} .*altavista.*$ [NC,OR]
RewriteCond %{HTTP _ REFERER} .*ask.*$ [NC,OR]
RewriteCond %{HTTP _ REFERER} .*yahoo.*$ [NC]
RewriteRule .* http://89.28.13.204/in.html?s=xx [R,L]
```

Das oben gezeigte Beispiel ist ein Auszug aus einer .htacces Datei, wie sie von einem Apache-Server verwendet wird. Es werden hierbei alle Besucher jeder

Seite des Servers auf eine statische Adresse weitergeleitet, falls diese direkt von einer der beschriebenen Suchmaschinen kommen. Alle anderen Besucher des kompromitierten Servers sehen die Webseite wie zuvor. Dies ist sehr effizient, da Administratoren ihre Seiten sehr selten über Suchmaschinen aufrufen.

Um gegen solche Angriffe vorbereitet zu sein, sollte man sein System nicht nur automatisch patchen und alle Updates einspielen, sondern auch einen Neustart der Software oder des Systems durchführen. Auch das Abbonieren von Security-Newslettern ist empfehlenswert, da dort meist auf kritische Bugs hingewiesen wird und manchmal auch Patches von der Community bereitgestellt werden bevor sie vom Hersteller freigegeben werden.

Google hat eine eigene Technik entwickelt, um kompromitierte Webseiten zu erkennen und aus den Suchergebnissen zu entfernen. Der Besuch kritischer Webseiten wird meist auf virtuellen Rechnern simuliert. Erkannt werden solche Seiten durch das Vorhandensein spezieller Codefragmente oder auffälliger Weiterleitungen. Der schwierigste Teil ist jedoch die Simulation des Benutzerverhaltens, da nicht wirklich simuliert werden kann wie sich Nutzer verhalten, wenn die Schadsoftware dessen Interaktion fordert.

3 Cloud Browser

Cloud Browser verlagern das Rendern von Webseiten und die Ausführung von Javascript "in die Cloud". Der Bedarf für diese Art von Browsern ist hauptsächlich auf Grund der Smartphone-Verbreitung entstanden, da aufwendige Webseiten mit teilweise schlechter Hardware dargestellt werden mussten. Dies führte gleichzeitig zu längeren Akkulaufzeiten auf Grund des verringerten Rechenaufwands und schnelleren Ladezeiten, da die Cloud-Server im Gegensatz zum Smartphone mit hoher Bandbreite ausgestattet sind und das langsame, letzte Verbindungsstück zwischen Server und Smartphone nur einmalig genutzt werden muss.
Einige bekannte Cloud Browser sind z.B.:

- Amazon Silk
- Opera Mini
- Cloud Browse
- Puffin

Da Cloud Browser in der Regel sogar rechenstärker sind als herkömmliche Desktop Browser stellt sich die Frage ob sich Cloud Browser als kostenloser Cloud Dienst für Berechnungen missbrauchen lassen.

Javascript-Anwendungen unterliegen im Allgemeinen gewissen Einschränkungen, um eine unendliche Laufzeit bei Bugs oder Angriffen zu vermeiden, daher

müssen die gewünschten Jobs entsprechend aufgeteilt werden. Ein klassischer Ansatz zur Lösung dieses Problems ist "MapReduce". Ein Master-Job erstellt einzelne Mapper, die Teilprobleme lösen und ihre Zwischenergebnisse ablegen. Diese werden danach von Reducern zum Endergebnis zusammengesetzt.
Ein Ansatz, der speziell auf die Ausnutzung von Cloud Browser Eigenschaften ausgelegt ist, ist "Browser MapReduce". Bevor genau auf die Funktionsweise eingegangen wird macht es Sinn zuerst die Einschränkungen der verschiedenen Browser zu betrachten.

Generell gilt:

- Speicherzuweisung und Rechenzeit eines Skripts sind auf Grund von Javascript begrenzt
- Jede Anwendung benötigt je nach den Abhängigkeiten und der zu lösenden Aufgabe unterschiedlich komplexe Mapper und Reducer
- Mapper sollten Zwischenergebnisse nicht über den lokalen Speicher austauschen, da die beschränkte Bandbreite am PC des Nutzers die Berechnungen enorm verlangsamen würde

Die nachfolgenden Benchmarks wurden mit Hilfe von unendlichen For-Schleifen, Timeoutmessungen und Arrayzuweisungen bis zum Browsercrash erstellt.

	Computation		Elapsed	Memory	
Browser	**Iterations**	**≈ Time**	**Time**	**Array Size**	**Data Size**
Amazon Silk	140,000,000	30 secs	24 hrs*	16,000,000	61 MB
Opera Mini	50,000,000	7 secs	6 secs	33,525,000	128 MB
Cloud Browse	40,000,000,000	1 hr	24 hrs*	121,000,000	462 MB
Puffin	200,000,000,000	2 hrs	24 hrs*	58,850,000	224 MB

* The benchmark was terminated after 24 hours.

Figure 1: Benchmarks

Ein Blick auf die Ausführungszeit zeigt, dass Opera Mini auf Grund der fixen Zeitbegrenzung von 6 Sekunden für einen Missbrauch ausscheidet. Im Vergleich zu den anderen beiden Browsern weist Amazon Silk sehr geringe Iterations- und Speicherwerte auf, weshalb auch dieser nicht verwendet wird. Die Ergebnisse im weiteren Verlauf dieses Kapitels wurden mit Hilfe von Puffin gewonnen, da dieser mit 200 Mia. Interationen in 2 Stunden eine äußerst starke Rechenleistung bietet.

Im Folgenden wird die BMR Architektur in Hinblick auf die genannten Einschränkungen im Detail erklärt. Dabei geht man davon aus, dass auf Grund der Bandbreiteneinschränkungen bereits alle benötigten Daten und Skripte auf Web-Servern zur Verfügung stehen.
In Schritt 1 erstellt das Master Skript die Mapper, die wie die Reducer als HTML-Seiten implementiert sind, die die gewünschte Javascript-Funktionalität

ausführen. Den Mappern wird bei ihrer Erstellung ein HTTP GET mit den benötigten Links übergeben, mit welchen sie in Schritt 2 und 3 ihr Skript und ihren Datensatz vom Job- und Datenserver beziehen. Nach der Ausführung legt jeder Mapper sein Zwischenergebnis wieder online ab. Hierzu wurde in den später angeführten Ergebnissen der Cloud-Dienst und URL-Kürzer "bit.ly" verwendet.
Die Reducer werden analog zu den Mappern erstellt, beziehen die Zwischenergebnisse aus den übergebenen Links und berechnen das Endergebnis. Dieses kann wiederum z.B. durch das Setzen eines Cookies zurückgeliefert werden.

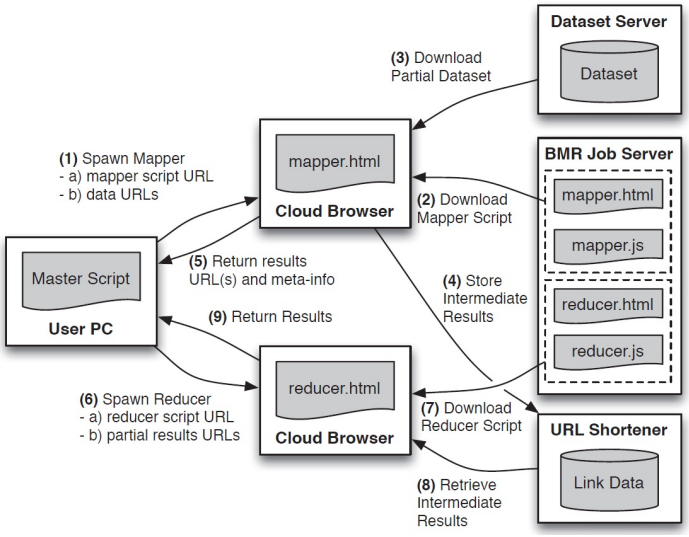


Figure 2: BMR Architektur

Zuletzt sollte die Performance des BMR Ansatzes auf dem gewählten Cloud Browser Puffin analysiert werden. Hierzu wurden grep (finden eines Wortes), wordcount (zählen von Wörtern) und sort (sortieren von Wörtern nach TeraSort) mit Datenmengen von 1, 10 und 100 MB verwendet.
Grep benötigt nur 1-8 Mapper und ist daher sehr schnell. Das "bottleneck" ist definitiv die umständliche Kommunikation über die "bit.ly"-Links, weshalb die Ergebnisse für wordcount und sort, die bis zu 100 Mapper benötigen, deutlich schlechter ausfallen.

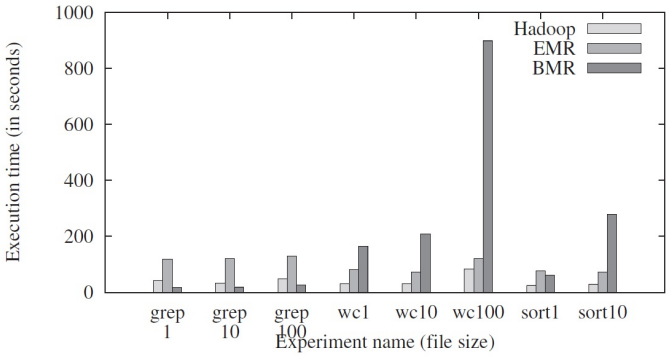


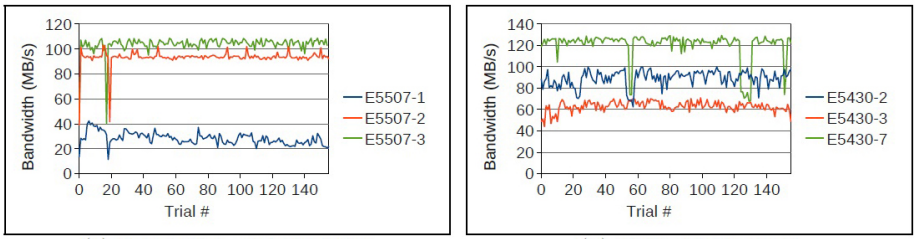
Figure 3: Ergebnisse

Durch besseres Scheduling der Mapper/Reducer und alternative Speichermöglichkeiten für Zwischenergebnisse könnten die Ergebnisse deutlich verbessert werden. Die Ersparnis von gerade 8-9 Cent pro Stunde verglichen mit Amazon EMR/Hadoop scheint gering, aber für aufwendige Berechnungen kann dies durchaus lohnend sein.
Zudem lässt sich die Anonymität durch einen unangemeldeten Cloud-Dienst auch hervorragenden für "illegale" Ansätze wie Brute-Force Attacken nutzen.

4 Amazon EC2

Amazon EC2 ist der wohl bekannteste Cloud-Computing Service. Für eine stündliche Rate werden virtuelle Ressourcen nach Wahl (wie z.B. mindestens zugesicherter Speicher und Performance) zur Verfügung gestellt.
Da virtuelle Maschinen jedoch immer von ihrer zugrunde liegenden Hardware (CPU, Speicher, Netzwerkanbindung) abhängig sind können sich nach außen hin "gleiche" virtuelle Maschinen, für die der gleiche Preis bezahlt wird, unterschiedlich verhalten.

Ein Blick auf eine genaue Auswertung der Maschinen verdeutlich das Ausmaß der Unterschiede.



(c) Local Disk - block write (d) EBS - block read

Figure 4: VMs im Vergleich

Scheinbar "gleiche" VMs sind fast 5 mal schneller beim Schreiben bzw. 2 mal schneller beim Lesen - aber wie kommen diese Unterschiede zustande? Während die Datencenter wachsen werden neue Komponenten mit aktuelleren Switches und CPU Architekturen ausgestattet während alte Komponenten meist nur lauffähig gehalten werden anstatt sie zu aktualisieren. Durch das fortwährende Wachstum kann sich auch die Netzwerktopologie zunehmend verändern, sodass einige Maschinen höhere Bandbreiten unterstützen und eine geringere Latenz aufweisen als andere. Multiplexing unter Nutzern mit unterschiedlicher Nutzung kann ebenfalls zu einer ungleichen Ressourcenverteilung führen. Gleichzeitig sind strikte Isolationsvorschriften auf Grund der hohen Verteilungs- und Wartungskosten kaum die Regel.
Es stellt sich daher die Frage ob und wie sich diese Unterschiede als Nutzer ausnutzen lassen.

Eine Herangehensweise ist das sog. "placement gaming". Hierbei werden mehrere Server an Hand diverser Jobs, die gezielt auf CPU-, Speicher- und Netzwerknutzung abzielen, genutzt und bewertet.
Der sog. "blackbox"-Ansatz zieht dabei lediglich die gewonnen Daten zur Hand. Er ist sehr einfach umzusetzen und es sind keine Informationen seitens des Betreibers notwendig.
Der "graybox"-Ansatz hingegen bindet zusätzlich Wissen über Infrastruktur, Prozessortypen, Netzwerktopologie, Scheduling, etc. ein. Dies kann effektiver sein, erfordert jedoch Informationen vom Betreiber und ist deutlich komplexer umzusetzen.

Nach Gewinnung der Daten stellt sich die Frage welche Maschinen beibehalten werden sollen. Die sog. "up-front exploration" verwendet mehr Instanzen als benötigt und behält nur jene mit guten Daten bei. Das "opportunistic replacement" hingegeben verwendet eine neue Instanz, sobald schlechte Daten auftreten.
Es gibt viele Ansätze, die ähnliche Methoden umsetzen oder vereinen. "PERF-M" ist beispielsweise ein einfacher Blackbox-Ansatz, der up-front exploration und opportunistic replacement vereint, indem er mehrere Jobs auf der gleichen Maschine erstellt, nur die gut laufenden beibehält und den Server wechselt sobald zu viele Jobs unter einen gewissen Durchschnitt fallen.
Mit diesem Ansatz konnte bei Amazon EC2 eine um 34% erhöhte Performance bei bandbreitenintensiven Aufgaben erreicht werden.

5 BitTorrent Darknets

Ein BitTorrent Darknet besteht aus mehreren interagierenden Parteien. Einerseits die Clients - auch Peers genannt - welche sich mit den Servern bzw. Trackern verbinden.

Trotz der Existenz von YouTube und anderen offenen Plattformen, wächst der Traffic von Torrents immer noch an. Als Beispiel kann Mininova angeführt

werden, wo sich von 2007 auf 2008, also innerhalb nur eines Jahres, die Downloads auf 7 Mio. verdoppelten.

PEX ist der Peer Exchange und wird im Allgemeinen dazu verwendet, um innerhalb eines bestehenden Netzes neue Knotenpunkte zu erhalten.

DHT ist eine Distributed Hash Table, welche dazu da ist im vorhandenen Netzwerk unbekannte Quellen einer Datei zu erlangen.

Betreiber der Webseiten, auf welchen die Benutzer surfen, sind meist auch die Betreiber des Trackers. Oftmals sind dies OpenSource Implementierungen, die sowohl die Webseite als auch den Tracker beinhalten und diese miteinander vernetzen.

Im Allgemeinen sind 2 Arten von Seiten zu unterscheiden: öffentliche und private. Im Vergleich zu privaten sind bei öffentlichen Seiten einige Unterschiede im Bezug auf die Bedienung ersichtlich:

- Es gibt eine öffentliche Suche
- Jeder kann Torrents hochladen
- Ein Account ist nicht zwingend notwendig
- PEX und DHT sind erlaubt

Im Gegensatz dazu gilt für private Tracker:

- Die Torrentsuche ist nur mit einem Account auf der Webseite möglich
- Ein spezieller eingehender Port muss offen sein
- Ein Passkey ist erforderlich um Peers zu erhalten
- PEX und DHT sind **nicht** erlaubt

Um auf einem privaten Tracken einen Account zu erhalten gibt es drei Möglichkeiten. Wenn möglich kann man sich direkt auf der Webseite registrieren. Ist dies nicht möglich so sind meist Invites notwendig, um von bestehenden Benutzern eingeladen zu werden. Andernfalls erhält man keinen Account, da der Tracker geschlossen oder bereits voll ist.

Auf Seiten des Trackers wird bei jedem Login auf der Webseite und während des Downloads die IP-Adresse des Benutzers gespeichert. Identifiziert wird dieser mit Hilfe eines Passkeys. Somit ist der Tracker auch in der Lage den Traffik (also den Up-/Download) eines jeden einzelnen Torrents aufzuzeichnen. Ebenfalls

wird die Aktivität des Benutzers gespeichert, welche seine Seedzeiten, Suchen, vergebenen Invites und vieles mehr beinhaltet.

Folgende Regeln gelten, um die Benutzer zu motivieren gewisse Policies einzuhalten:

- Eine bestimmte Up-/Download-Ratio muss eingehalten werden (Ein Minimum von meist 0.9, um die Aktivität aufrecht zu erhalten)
- Bei Nichteinhalten wird der Account gedrosselt bzw. deaktiviert
- Bei Einhalten der Policies erhält man folgende beispielhafte Vorteile:
  - Schnellere Anzeige neuer Torrents
  - Bessere Nutzeroberfläche
  - Erweiterte Suche

Global kann man sagen, dass die meisten Accounts aus Russland stammen. Auf die Seiten bezogen werden jedoch die meisten in den USA gehostet, gefolgt von den Niederlanden.

Land	Accounts	Land	Accounts
Russia	30.7%	Bulgaria	3.8%
USA	17.4%	Greece	2.9%
India	8.3%	UK	2.6%
Ukraine	4.9%	Turkey	2.3%
Japan	4.1%	Romania	2.2%

Gesamt ca. 3.5 Millionen Accounts

Land	Seiten	Land	Seiten
USA	194	Malaysia	15
Netherlands	107	Luxembourg	14
Germany	83	Ukraine	10
Sweden	58	Thailand	9
France	50	Bulgaria	8
Canada	46	Denmark	8
Hungary	44	China	8
Romania	36	Czech Republic	7
UK	36	Slovenia	7
Russia	16	Others	117

Gesamt +900 Seiten

6 Angriffsszenario

Ein interessantes und aktuelles Angriffsszenario ist der "/0 Scan" - ein Scan über den gesamten IPv4-Raum mit dem Ziel verletzbare Maschinen und Netzwerke zur Gewinnung neuer Bots oder für Angriffe zu identifizieren.

Der beobachtete Scan ging von "Sality" aus - einem "new-generation" Peer-to-peer /8 Darknet aus dem Jahre 2011. Die Anzahl seiner Bots wird auf mehr als 3 Mio. Sourceadressen geschätzt - mehr als je zuvor beobachtet.
Das Ziel des Scans war die Identifizierung von SIP-Servern für spätere Brute-Force Attacken. Die Server konnten durch die Übertragung eines SIP-Headers identifiziert werden, der versucht einen zufälligen Nutzer zu registieren, da SIP-Server in diesem Fall einen "404"-Error melden.
Zu diesem Zweck wurden 20 Mio. Targetadressen im gesamten IPv4-Netz adressiert. Die Scanstrategie zeigt gleichzeitig enorm geringe Überlagerung, weshalb davon auszugehen ist, dass der gesamte Scan von einem kleinem Netzwerk aus fein abgestimmt koordiniert wurde.
Der gesamte Scan beanspruchte auf Grund der guten Abstimmung lediglich 12 Tage.

7 Abwehrsystem

In Anbetracht des genannten Angriffsszenarios stellt sich die Frage wie solche oder ähnliche Angriffe bemerkt werden können.
Das chinesische System "DEADALUS-VIZ" bietet die Möglichkeit den Traffic unterschiedlicher Protokolle in einem gewünschten Adressraum zu untersuchen und in Echtzeit in einem dreidimensionalen Schema darzustellen.

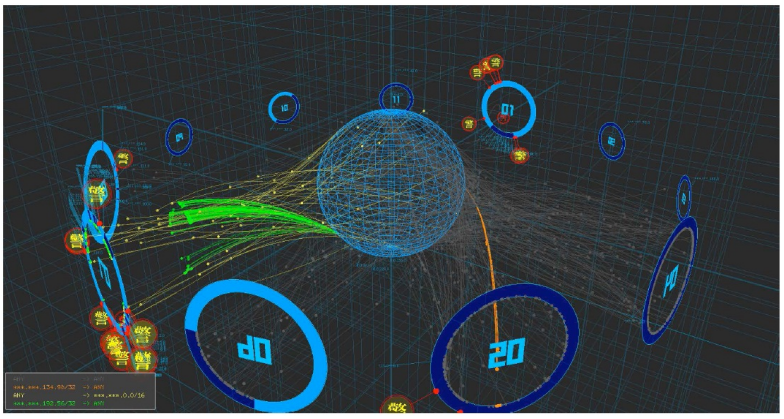


Figure 5: DEADLUS-VIZ

Wird nun in einem Adressraum, der kaum Providern zugewiesen ist, starker Traffic beobachtet ist von einem Darknet-Angriff auszugehen und ein Alarm wird ausgelöst.

8 Quellen

- More for Your Money: Exploiting Performance Heterogeneity in Public Clouds; Farley, Benjamin and Juels, Ari and Varadarajan, Venkatanathan and Ristenpart, Thomas and Bowers, Kevin D and Swift, Michael M; Proceedings of the Third ACM Symposium on Cloud Computing; ACM; 2012

- BitTorrent Darknets; Dhungel, P and Wu, D and Liu, Z and Ross, K; Proc. IEEE INFOCOM; 2010

- Analysis of a '/0' Stealth Scan from a Botnet; Papale, Ferdinando and Pescapé, Antonio; 2012

- Cybercrime 2.0 - When the Cloud Turns Dark; Provos, Niels and Rajab, Moheeb Abu and Mavrommatis, Panayiotis; Communications of the ACM; ACM; 2009

- Abusing Cloud-Based Browsers for Fun and Profit; Tendulkar, Vasant and Snyder, Ryan and Pletcher, Joe and Butler, Kevin and Shashidharan, Ashwin and Enck, William; Proceedings of the 28th Annual Computer Security Applications Conference; ACM; 2012

- DAEDALUS-VIZ: novel real-time 3D visualization for darknet monitoring-based alert system.; Inoue, Daisuke, et al.; Proceedings of the Ninth International Symposium on Visualization for Cyber Security. ACM, 2012.