RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: EXP-W14

HACKING EXPOSED: THINK BEYOND (NEXTGEN AI POWERED)

Stuart McClure

CEO, Co-Founder
Cylance Inc.

Paul Mehta

Chief Architect
Cylance Inc.

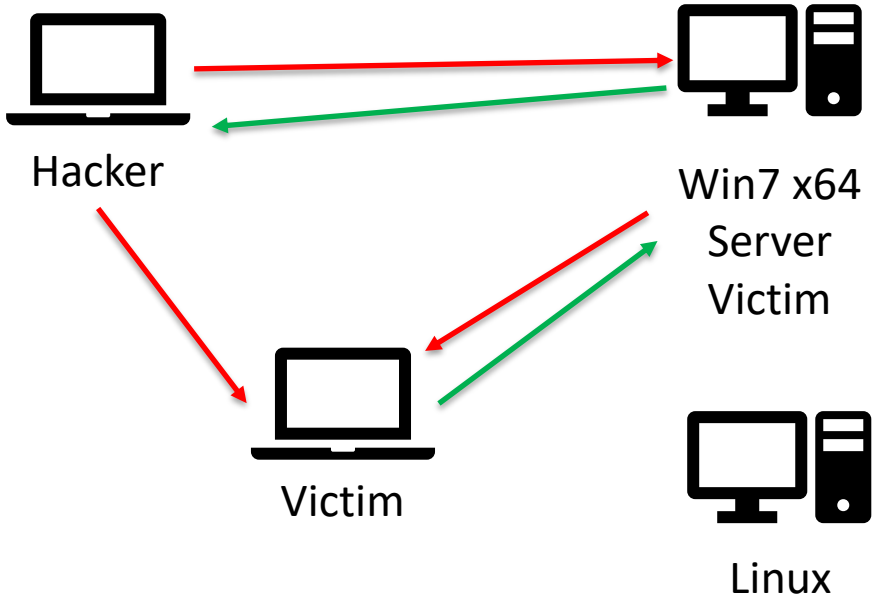
Brian Wallace

Lead Security Data Scientist
Cylance Inc.

Agenda

- Network hack: Server
  - EternalBlue->Doublepulsar
  - Password dumping the Server
    - Update->Dump Credentials
- Drive-by download: Spectre and Meltdown
  - Spectre (User) – Demo on Windows
    - Dump memory from a scripting language (without reading memory)
  - Meltdown (Kernel) – Demo on Linux
    - The cure, worst than the disease on Windows 7 x64
- SpearPhish: Operation Shaheen
  - Email->DOC->Crypto attack (with an unusual twist)

Hack Map



HACK #1

SERVER OVER NETWORK HACK

ETERNELBLUE->DOUBLEPULSAR

Is the backdoor locked?


```
.data:000000000001D75D                db 0FEh ; þ
.data:000000000001D75E                db 0FEh ; þ
.data:000000000001D75F                db 0FEh ; þ
.data:000000000001D760 SrvTransaction2DispatchTable dq offset SrvSmbOpen2
.data:000000000001D760                                        ; DATA XREF: ExecuteTransaction+BE↓r
.data:000000000001D768                dq offset SrvSmbFindFirst2
.data:000000000001D770                dq offset SrvSmbFindNext2
.data:000000000001D778                dq offset SrvSmbQueryFsInformation
.data:000000000001D780                dq offset SrvSmbSetFsInformation
.data:000000000001D788                dq offset SrvSmbQueryPathInformation
.data:000000000001D790                dq offset SrvSmbSetPathInformation
.data:000000000001D798                dq offset SrvSmbQueryFileInformation
.data:000000000001D7A0                dq offset SrvSmbSetFileInformation
.data:000000000001D7A8                dq offset SrvSmbFindNotify
.data:000000000001D7B0                dq offset SrvSmbIoctl2
.data:000000000001D7B8                dq offset SrvSmbFindNotify
.data:000000000001D7C0                dq offset SrvSmbFindNotify
.data:000000000001D7C8                dq offset SrvSmbCreateDirectory2
.data:000000000001D7D0                dq offset SrvTransactionNotImplemented
.data:000000000001D7D8                dq offset SrvTransactionNotImplemented
.data:000000000001D7E0                dq offset SrvSmbGetDfsReferral
.data:000000000001D7E8                dq offset SrvSmbReportDfsInconsistency
.data:000000000001D7F0 SrvNtTransactionDispatchTable dq 0      ; DATA XREF: ExecuteTransaction+1FB↓r
.data:000000000001D7F8                dq offset SrvSmbCreateWithSdOrEa
.data:000000000001D800                dq offset SrvSmbNtIoctl
.data:000000000001D808                dq offset SrvSmbSetSecurityDescriptor
.data:000000000001D810                dq offset SrvSmbNtNotifyChange
.data:000000000001D818                dq offset SrvSmbNtRename
.data:000000000001D820                dq offset SrvSmbQuerySecurityDescriptor
.data:000000000001D828                dq offset SrvSmbQueryQuota
.data:000000000001D830                dq offset SrvSmbSetQuota
.data:000000000001D838                dq offset SrvSmbCreateWithExtraOptions
.data:000000000001D840 SrvApiDispatchTable dq 0                ; DATA XREF: sub_68FC4+B9↓r
.data:000000000001D848                db    0
.data:000000000001D849                db    0
```

CYLANCE

Hack Steps

- Windows 7 x64 server
- Vulnerable SMBv1 running
- EternalBlue (the exploit) delivering Doublepulsar (the payload)
- Controlled by fb.py, FuzzBunch, an NSA Tool for exploits and implants
- Upload files as part of a campaign

HACK #1
DEMO

(praise to the live demo gods!)

Preventing EternelBlue

- Patching is about it…

HACK #2
DRIVE-BY DOWNLOAD
SPECTRE AND MELTDOWN

Global Meltdown

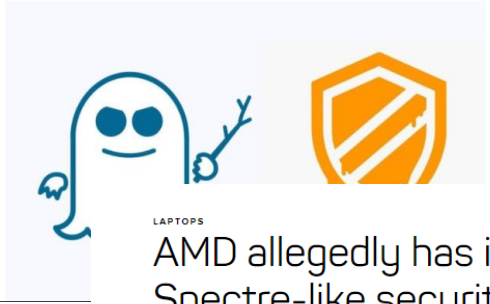
UPDATE

Meltdown and Spectre FAQ: How the critical CPU flaws affect PCs and Macs

It varies widely depending on your hardware, operating system, and workload.

By Brad Chacos and Michael Simon

PCWorld | FEB 22, 2018 7:14 AM PT



MORE LIKE THIS


How to protect your PC from the Meltdown and Spectre CPU flaws


AMD processors will get a firmware update for the Spectre CPU exploit, but...


Nvidia updates graphics

LAPTOPS

AMD allegedly has its own Spectre-like security flaws

Researchers say they've found 13 flaws in AMD's Ryzen and EPYC chips, which could let attackers install malware on highly guarded parts of the processor.

AMD Releases Updated Risk Guidance on Meltdown, Spectre

By Joel Hruska on January 11, 2018 at 5:39 pm | 41 Comments

64 SHARES

With Meltdown and Spectre now unveile
contain the problems they represent anc
Different vendors have released their ov
Meltdown, ARM has some limited vulner
quiet, apart from its initial statement last

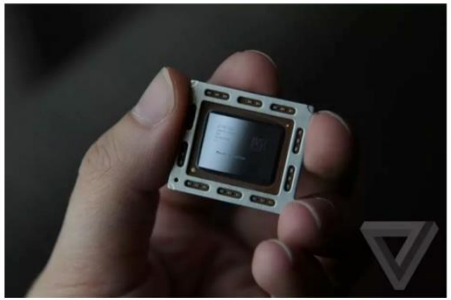
MICROSOFT | TECH

Microsoft halts AMD Meltdown and Spectre patches after reports of unbootable PCs

Microsoft blames AMD's documentation

By Tom Warren | @tomwarren | Jan 9, 2018, 2:50am EST

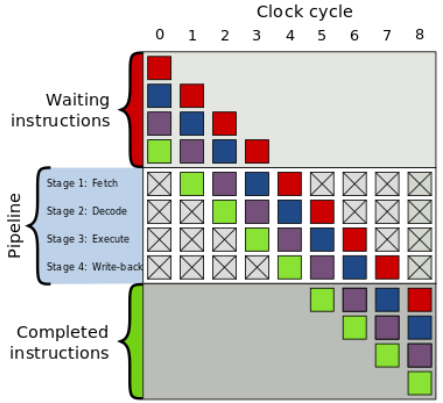
SHARE



CYLANCE

Instruction Pipelines

- One of the ways that modern processors increase speed is by pipelining
- Pipelining allows instructions to execute faster and in parallel
- What about branch instructions?
  - This is where Speculative Execution comes in



Speculative execution

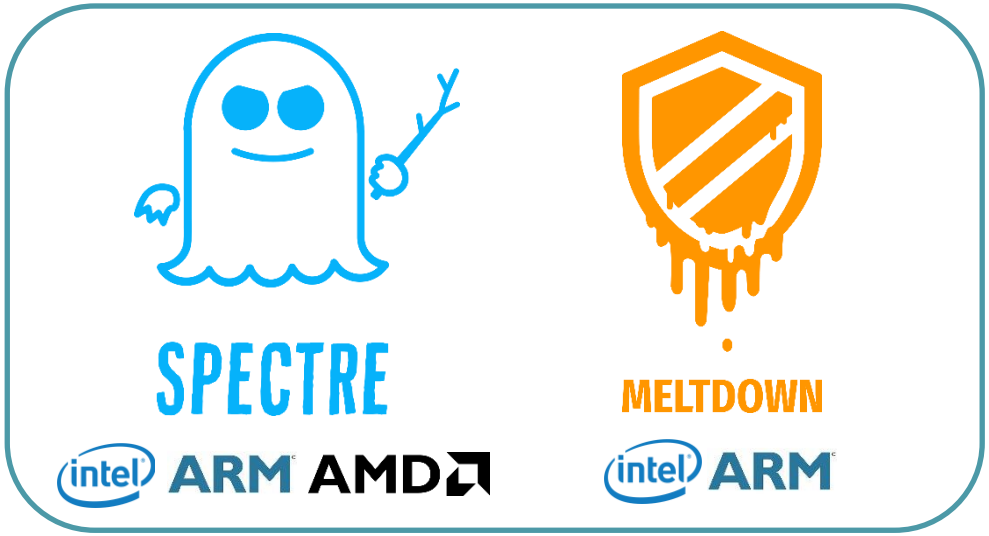
- A vulnerable processor encounters a **branch**, if it's really not sure which branch to take…
- The **Branch** is *both taken and not taken*.
- Once determined, the other branch is discarded.
- Observe side effects from the discarded work. Wait, what?
  - *The branch not taken is discarded, so how can the effects be observed?*

The Crux

- The effects of touching memory will remain in the CPU cache.
- Cache hits are faster than misses.
- By measuring memory access this time, an attacker can infer memory.
- Repeating many times increases reliability.

Hack Steps

- Windows 7 desktop
- Vulnerable Windows and Linux version running
- User goes to website
  - Run Spectre
  - Show User memory
- User runs meltdown
  - Show Kernel memory

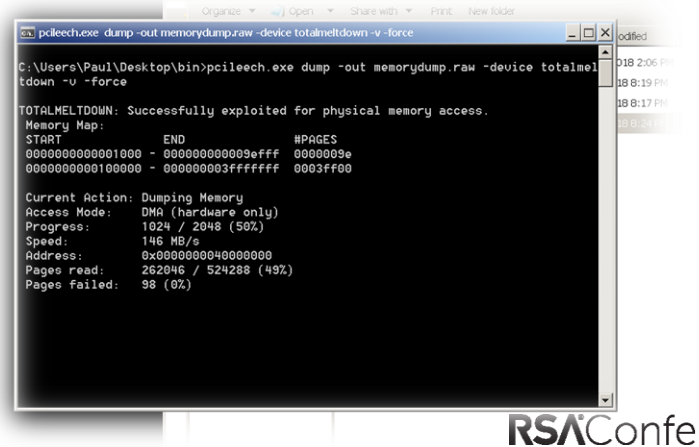


HACK #2
DEMO: SPECTRE AND MELTDOWN

(praise to the live demo gods!)

Total meltdown!

- On Windows 7 x64, this patch introduced a larger security hole...
  - The user/supervisor bit was set to user
  - Allowed user-mode code access to the page tables… oops
- This means userland code can access page tables and dump kernel memory with ease at 100MB+/s



Prevention: Meltdown Patch…….?

- Microsoft released a patch for Meltdown in January
- Kb4056897 implemented “Kernel Page Table Isolation”, or KPTI to mitigate Meltdown
  - On every context switch, page tables are swapped back and forth between kernel and userland.
  - *As you know*, page tables map virtual memory to physical memory.
  - Patch does not prevent Meltdown from leaking memory so much as it prevents the memory from being present.
- This was fixed in the March patch

HACK #3
DEMO: CURE WORSE THAN THE DISEASE?

(praise to the live demo gods!)

HACK #4
SPEARPHISHING OPERATION SHAHEEN

APT group attacking Pakistani Nuclear Power Plant

SpearPhishing: Operation Shaheen

- We were looking for similar documents related to an APT from another group.

- Led us to a hacked Belgian Company site.
  - The hacked site was being used as a platform to distribute exploits.

Exploit Document

- The exploit contained dl/exec shellcode which pulled a payload from the Government website.

- The military engineering organization, and one of the major science and technology commands of a nation state Army.

Document attack

- This document was downloaded from an IP address
- C&C domain resolution using web services and attacks against Asian targets.
- Which for a period of a week resolved to this IP: 1**.2**.1**.1**.
  - resolved to a subdomain of a completely unrelated topic.

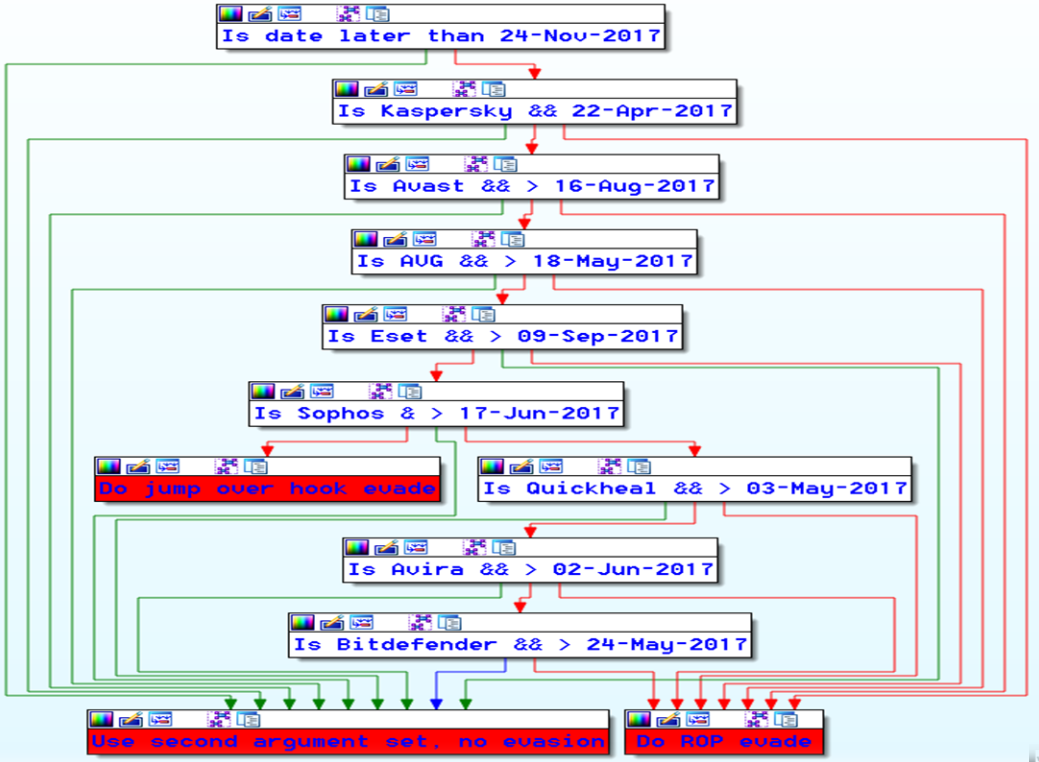
Evolution

- The shellcode has evolved over time.
  - Earlier exploit docs contained basic download & execute shellcode.
  - More recent examples use multi-stage embedded payloads.
- Our samples matched based on source of the payload as well as shellcode similarities, known as code-sharing.

Evasion Capabilities

- They used a relatively unique means of obfuscating function calls to evade Kaspersky.
- They went through a lot of trouble to do this among other evasions, only to stop in sync.

Evasion: Antivirus?



HACK #4
DEMO: SPEARPHISH SHAHEEN

(praise to the live demo gods!)

Prevention: Operation Shaheen

- Office 2010 SP2 patch, July 2013

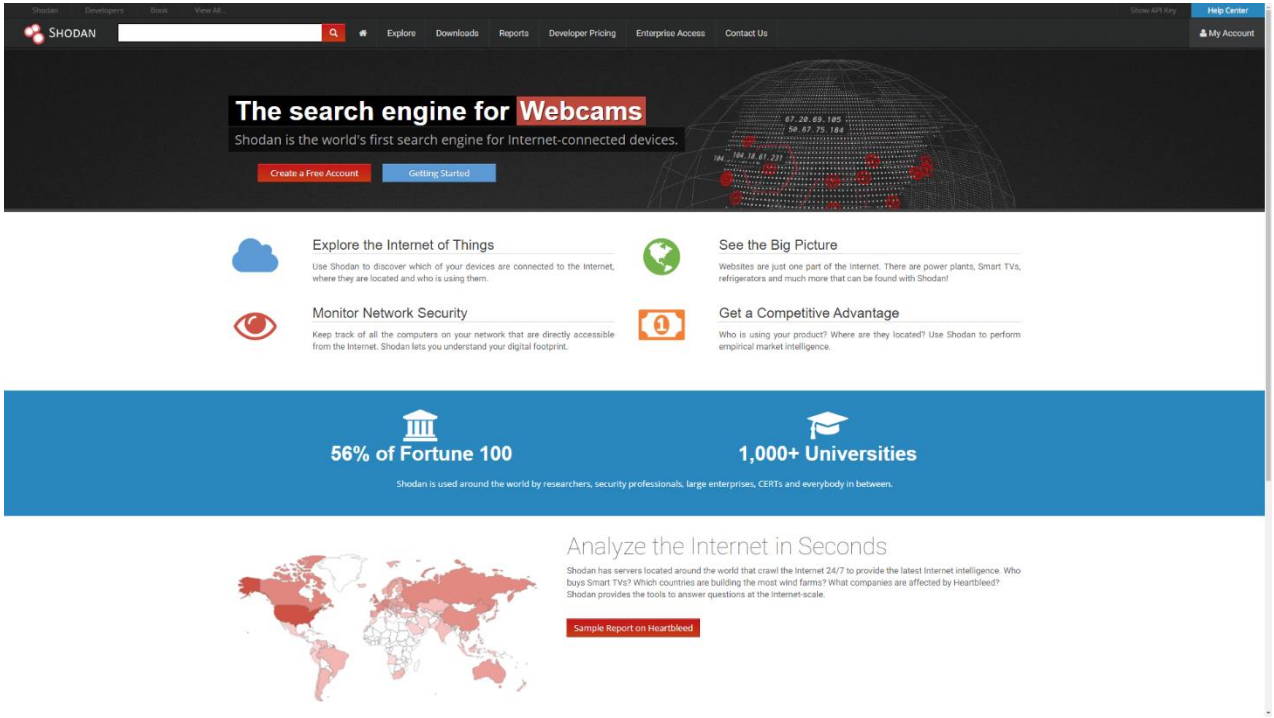
CYLANCE

HACK #5
ADVERSARIAL AI

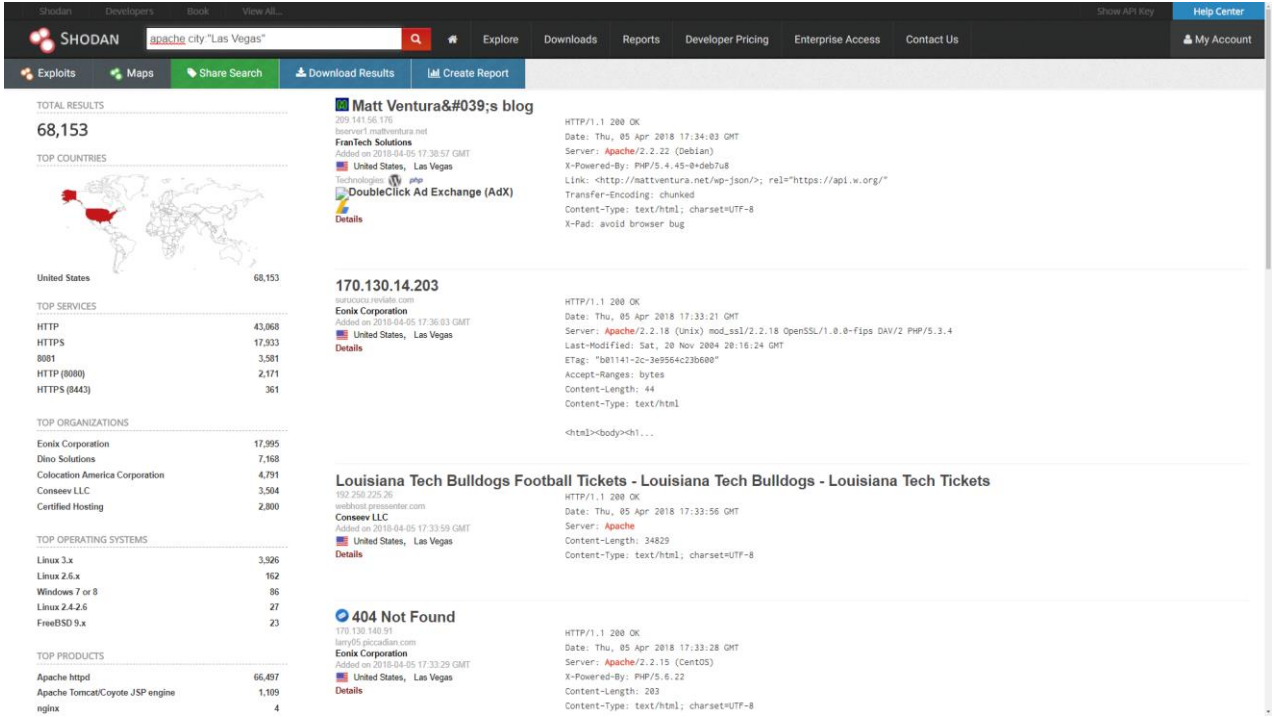
Adversarial AI

- The practice of pitting AI against AI to improve the original classification mode
- Build classification model
- Search for Interesting Results
- Build adversarial generated deceptive samples to trick our model into missing those results
- Harden the model against attacks by training it with samples generated to attack it

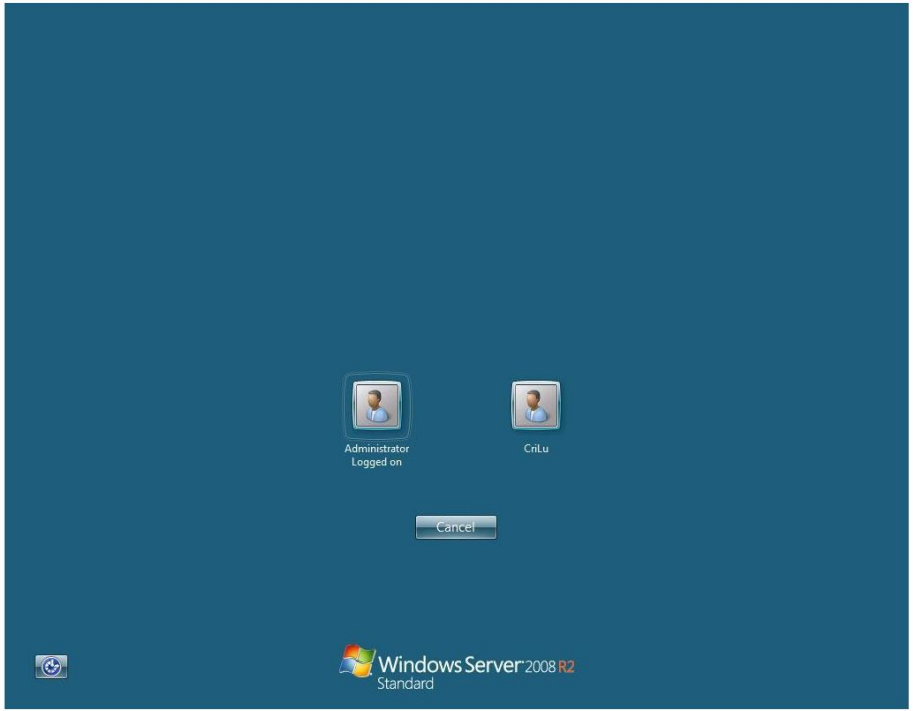
Shodan – Scans/Archives public IP space



Shodan - Queries



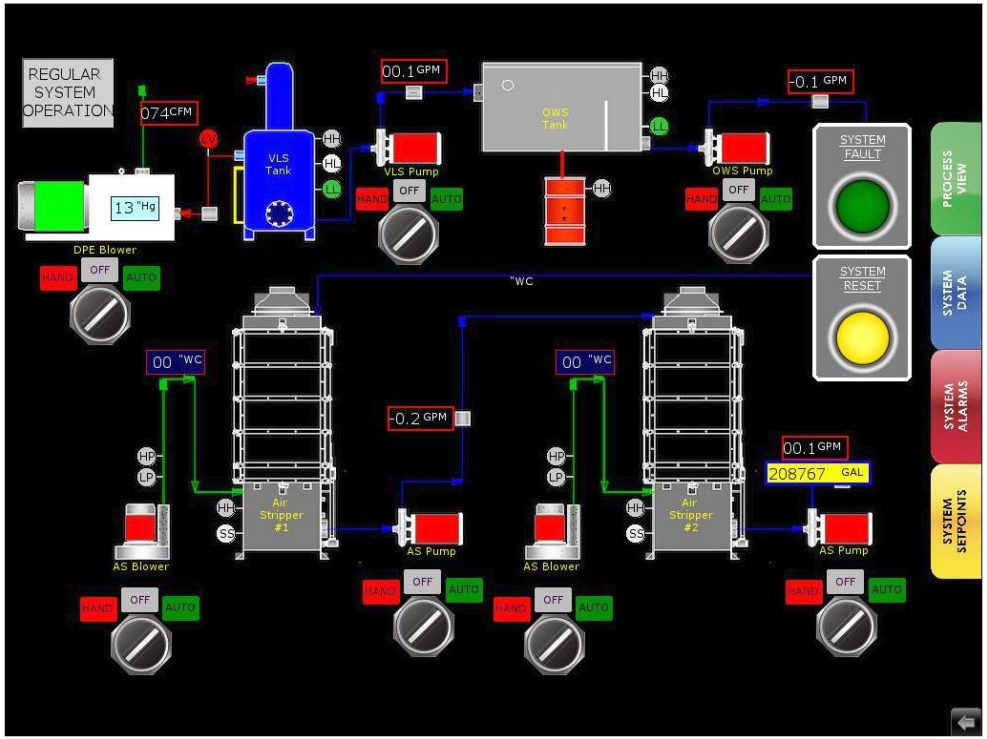
Shodan Grabs Screenshots



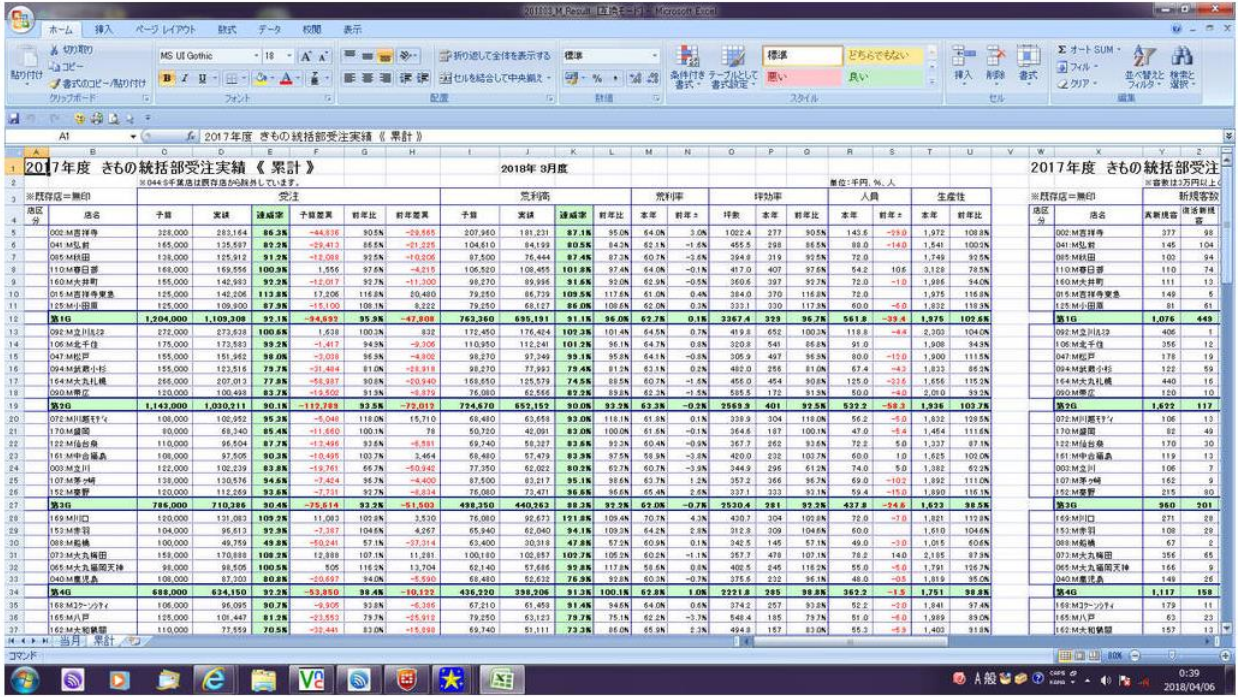
Shodan Sees Security Cams

Shodan Sees POS

Shodan Sees ICS



Shodan Sees Logged in Desktops



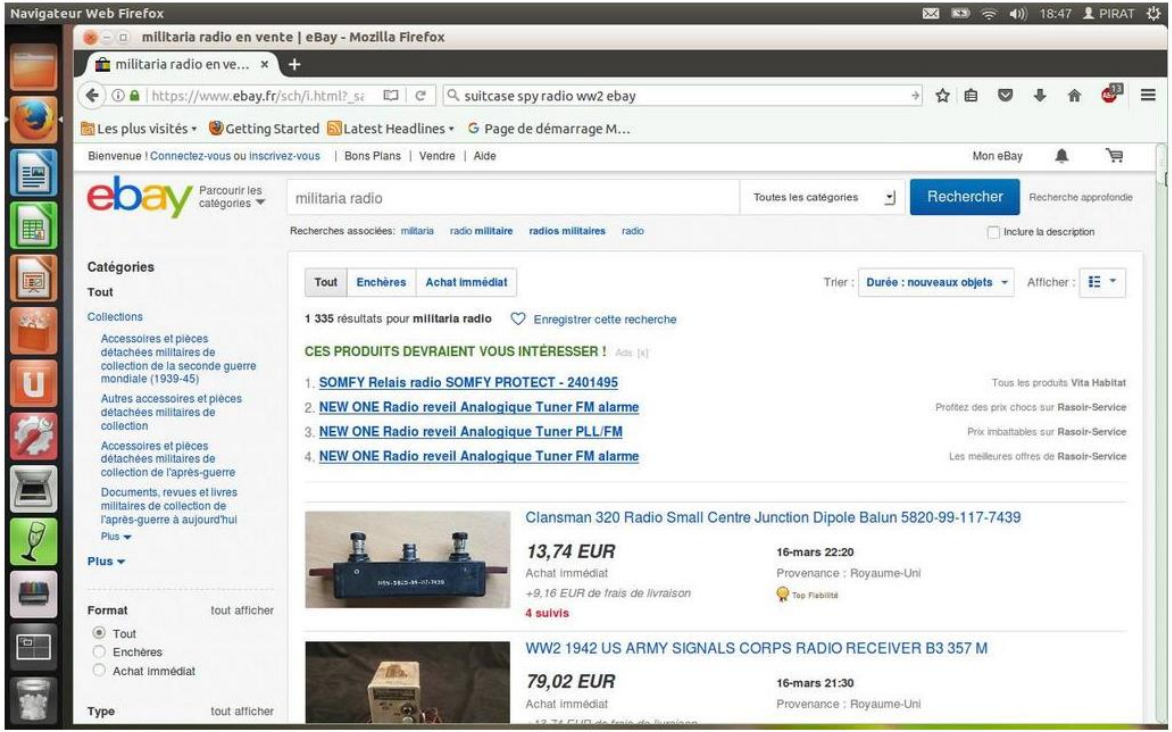
2017年度 きもの統括部受注実績《 累計 》

2018年 3月度

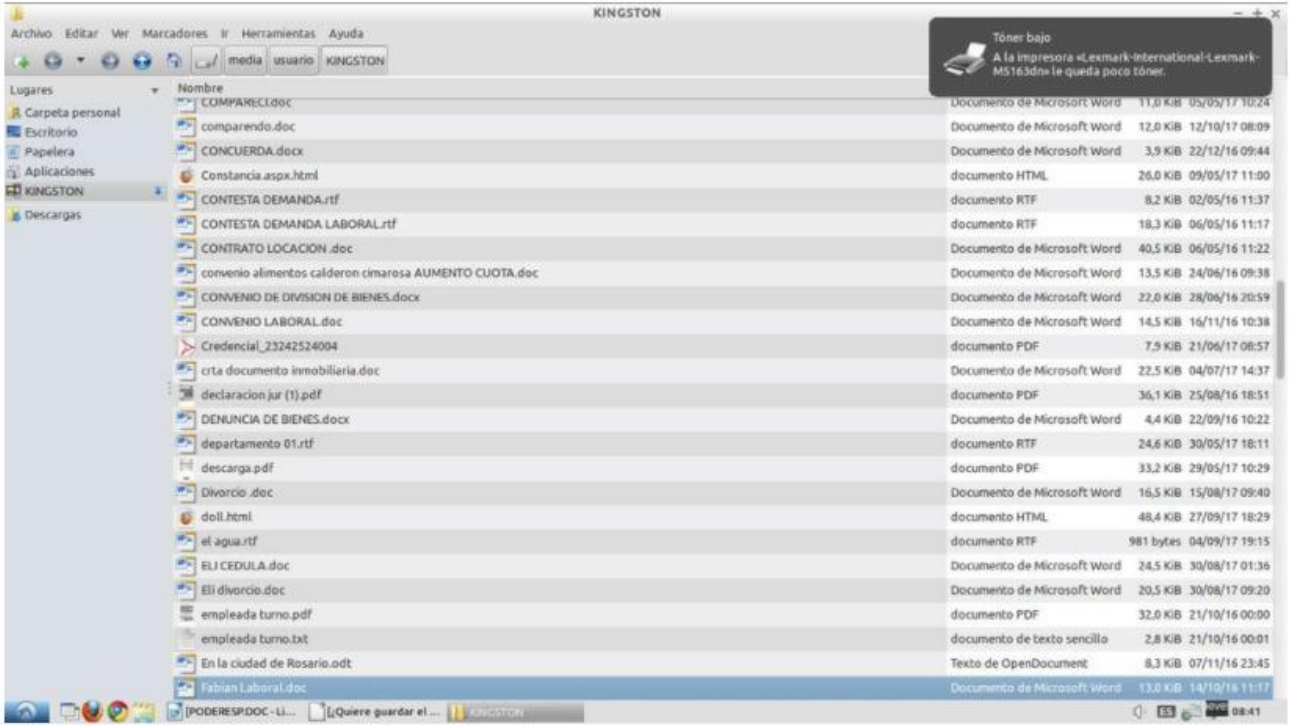
2017年度 きもの統括部受注

店区分	店名	受注 予算	受注 実績	受注 達成率	受注 予算差異	受注 前年比	受注 前年差異	荒利高 予算	荒利高 実績	荒利高 達成率	荒利高 前年比	荒利率 本年	荒利率 前年±	稼動率 稼動	稼動率 本年	稼動率 前年比	人員 本年	人員 前年±	生産性 本年	生産性 前年比	新規客数 真新規客	新規客数 復活新規客
	002:M吉祥寺	328,000	283,164	86.3%	-44,836	90.5%	-29,565	207,960	181,231	87.1%	95.0%	64.0%	3.0%	1022.4	277	90.5%	143.6	-19.0	1,972	108.8%	377	98
	041:M弘前	165,000	135,597	82.2%	-29,413	86.5%	-21,225	104,610	84,199	80.5%	84.3%	62.1%	-1.6%	455.5	298	86.5%	88.0	-14.0	1,541	100.2%	145	104
	065:M秋田	138,000	125,912	91.3%	-12,088	92.5%	-10,026	87,500	76,444	87.4%	87.3%	60.7%	-3.6%	394.8	319	92.5%	72.0		1,749	92.5%	103	94
	110:M春日部	169,000	169,556	100.3%	1,556	97.6%	-4,215	106,920	108,455	101.4%	97.4%	64.0%	-0.1%	417.0	407	97.6%	54.2	10.5	3,128	78.5%	110	74
	160:M大井町	155,000	142,983	92.2%	-12,017	92.7%	-11,300	98,270	89,996	91.6%	92.0%	62.9%	-0.5%	360.6	397	92.7%	72.0	-1.0	1,986	94.0%	111	13
	015:M吉祥寺東急	125,000	142,206	113.8%	17,206	116.8%	20,480	79,250	86,739	109.5%	117.6%	61.0%	0.4%	384.0	370	116.8%	72.0		1,975	116.8%	149	5
	125:M小田原	125,000	109,900	87.9%	-15,100	108.1%	8,222	79,250	68,127	86.0%	108.6%	62.0%	0.3%	333.1	330	117.3%	60.0	-6.0	1,832	118.3%	81	61
	第1G	1,204,000	1,109,308	92.1%	-94,692	95.3%	-47,808	762,360	695,191	91.1%	96.0%	62.7%	0.1%	3367.4	329	96.7%	561.8	-39.4	1,975	102.6%	1,076	449
	092:M立川&2	272,000	273,628	100.6%	1,628	100.3%	832	172,450	176,424	102.3%	101.4%	64.5%	0.7%	419.8	652	100.3%	118.8	-4.4	2,303	104.0%	406	1
	106:M北千住	175,000	173,583	99.2%	-1,417	94.5%	-9,306	110,950	112,241	101.2%	96.1%	64.7%	0.5%	320.8	541	86.8%	91.0		1,908	94.3%	356	12
	047:M松戸	155,000	151,962	98.0%	-3,038	96.3%	-4,800	98,270	97,349	99.1%	95.8%	64.1%	-0.8%	305.9	497	96.3%	80.0	-12.0	1,900	111.5%	178	19
	094:M武蔵小杉	155,000	123,516	79.7%	-31,484	81.0%	-18,918	98,270	77,993	79.4%	81.3%	63.1%	0.3%	482.0	256	81.0%	67.4	-4.3	1,833	86.3%	122	59
	164:M大丸札幌	265,000	207,013	77.8%	-55,987	90.8%	-20,940	168,650	125,579	74.5%	88.5%	60.7%	-1.6%	455.0	454	90.8%	125.0	-23.6	1,656	115.2%	440	16
	090:M幕張	120,000	100,488	83.7%	-19,500	91.3%	-8,879	76,080	62,556	82.2%	89.8%	62.3%	-1.5%	585.5	172	91.3%	50.0	-4.0	2,010	99.2%	120	10
	第2G	1,143,000	1,030,211	90.1%	-112,789	92.5%	-72,013	724,670	652,152	90.0%	93.2%	63.3%	-0.3%	2569.9	401	92.5%	532.2	-58.3	1,936	103.7%	1,622	117
	072:M川越モディ	108,000	102,952	95.3%	-5,048	118.0%	15,710	68,480	63,658	93.0%	118.1%	61.8%	0.1%	339.9	304	118.0%	56.2	-5.0	1,832	128.5%	106	13
	170:M鎌岡	80,000	68,340	85.4%	-11,660	100.1%	78	50,720	42,091	83.0%	100.0%	61.5%	-0.1%	364.6	187	100.1%	47.0	-5.4	1,454	111.6%	82	49
	122:M仙台泉	110,000	96,504	87.7%	-13,496	93.6%	-6,881	69,740	59,327	85.1%	93.3%	60.4%	-0.9%	367.7	262	93.6%	72.2	5.0	1,337	97.1%	170	30
	161:M中古福島	108,000	97,505	90.3%	-10,495	103.7%	3,464	69,480	57,473	82.7%	97.5%	58.9%	-3.8%	420.0	232	103.7%	60.0	1.0	1,625	100.0%	119	13
	003:M立川	122,000	102,239	83.8%	-19,761	66.7%	-50,942	77,350	62,022	80.2%	62.7%	60.7%	-3.9%	344.9	296	61.2%	74.0	5.0	1,382	62.2%	106	7
	107:M茅ヶ崎	128,000	120,576	94.6%	-7,424	95.7%	-4,400	87,500	83,217	95.1%	98.6%	63.7%	1.3%	357.2	366	96.7%	69.0	-10.2	1,892	111.0%	163	9
	152:M愛甲	120,000	112,269	93.6%	-7,731	92.7%	-8,834	76,080	72,471	95.5%	96.6%	65.4%	2.6%	337.1	333	92.1%	59.4	-15.0	1,890	116.1%	215	80
	第3G	786,000	710,386	90.4%	-75,614	93.3%	-51,503	498,350	440,263	88.3%	93.2%	62.0%	-0.7%	2530.4	981	92.2%	437.8	-24.6	1,622	98.5%	960	201
	169:M川口	120,000	131,083	109.2%	11,083	102.8%	3,530	76,080	92,673	121.8%	109.4%	70.7%	4.3%	430.7	304	102.8%	72.0	-7.0	1,821	112.8%	271	28
	153:M赤羽	104,000	95,613	92.3%	-7,387	104.6%	4,267	65,940	62,040	94.1%	109.3%	64.2%	2.8%	312.8	309	104.6%	60.0		1,610	104.6%	108	28
	088:M船橋	100,000	49,759	49.8%	-50,241	57.1%	-27,314	63,400	30,318	47.8%	57.2%	60.9%	0.1%	342.5	145	57.1%	49.0	-3.0	1,015	60.6%	67	2
	073:M大丸梅田	158,000	170,888	108.2%	12,888	107.1%	11,281	100,180	100,857	100.7%	105.2%	60.2%	-1.1%	357.7	478	107.1%	78.2	14.0	2,185	97.3%	356	65
	065:M大丸福岡天神	98,000	98,505	100.5%	505	116.3%	13,704	62,140	57,686	92.8%	117.8%	58.6%	0.8%	402.5	245	116.3%	55.0	-5.0	1,791	126.7%	166	9
	040:M鹿児島	108,000	97,300	90.8%	-10,697	94.0%	-6,590	68,480	62,632	76.2%	92.8%	60.3%	-0.7%	375.5	232	95.1%	48.0	-0.5	1,819	95.0%	140	26
	第4G	688,000	634,150	92.2%	-53,850	98.4%	-10,122	436,220	398,206	91.3%	100.1%	62.8%	1.0%	2221.8	989	98.8%	362.2	-1.5	1,751	98.8%	1,117	158
	168:M3ターンシティ	106,000	96,095	90.7%	-9,905	93.8%	-6,386	67,210	61,459	91.4%	94.6%	64.0%	0.6%	374.2	257	93.8%	52.2	-2.0	1,841	97.4%	179	11
	165:M八戸	125,000	101,447	81.2%	-23,553	79.7%	-25,912	79,250	63,123	79.7%	75.1%	62.2%	-3.7%	548.4	185	79.7%	51.0	-8.0	1,989	89.0%	63	23
	162:M大和西口	110,000	77,559	70.5%	-32,441	83.0%	-15,998	69,740	51,111	73.3%	86.0%	65.9%	2.3%	494.8	157	83.0%	55.3	-5.3	1,403	91.8%	157	13

Shodan Sees Logged in Desktops



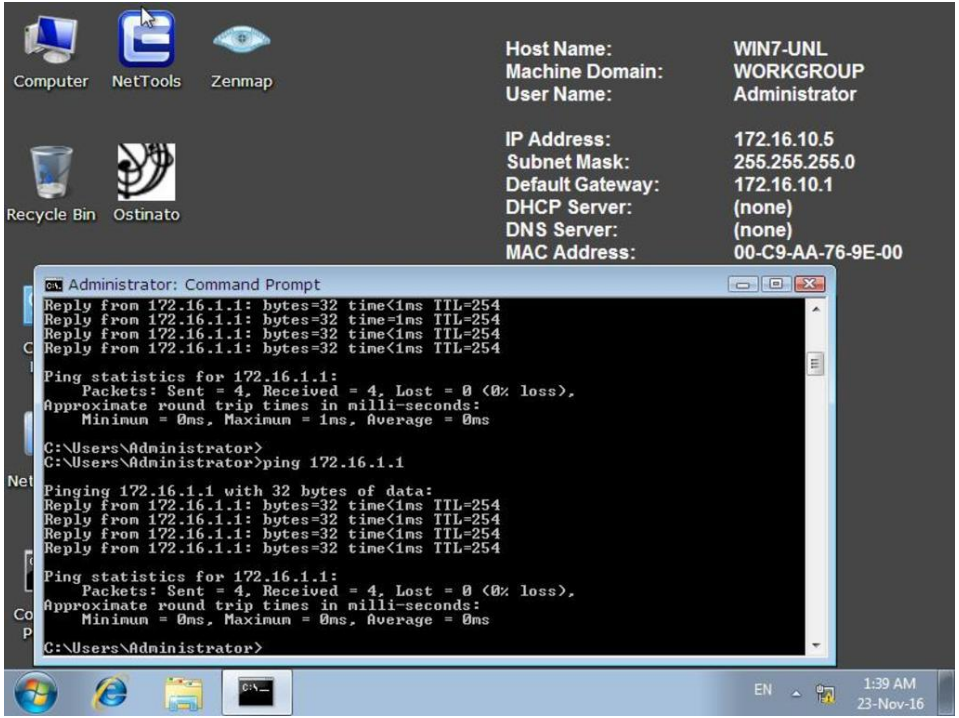
Shodan Sees Logged in Desktops



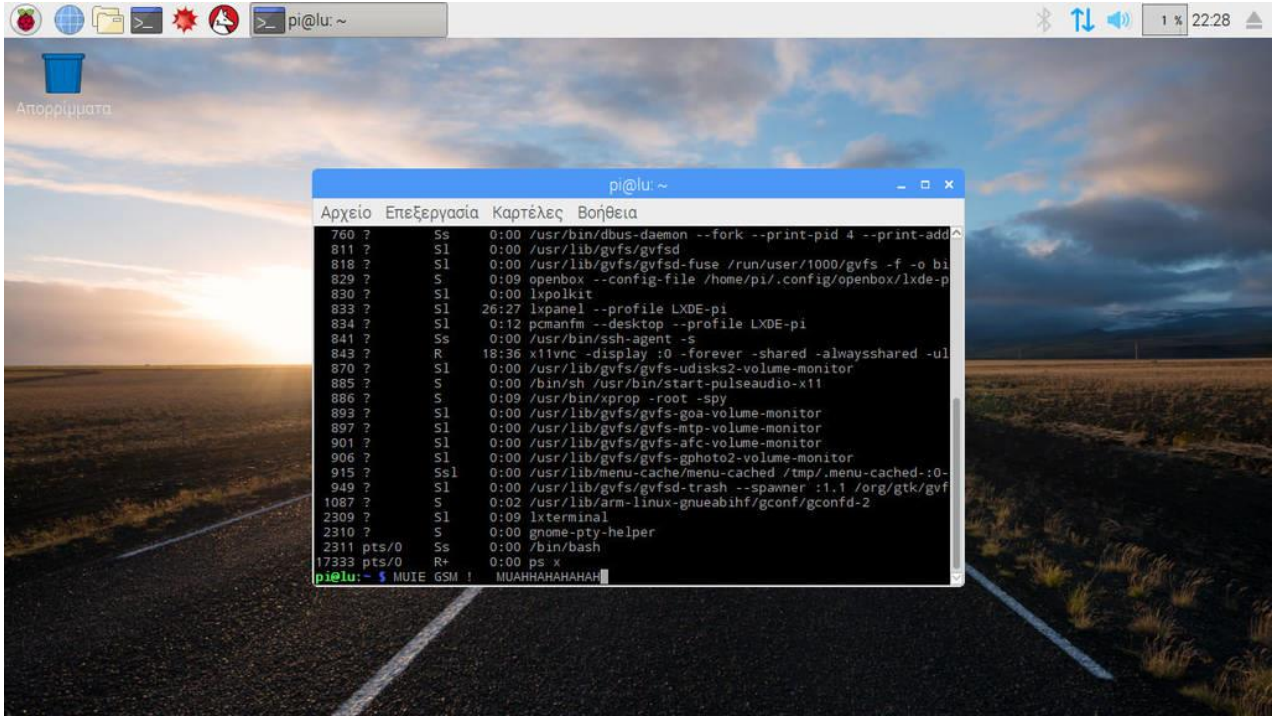
Shodan Image Classification (SIC) Project

- Objective: Build a collection of binary classifiers to identify different traits about screen shots of Shodan services
  - Including classifiers that determine if a screen shot is a logged in desktop, ICS, POS, log in screen, security camera, etc
- Downloaded and manually labeled as many images as I could from the Shodan Developer API
- Trained two part Neural Networks using a method called “Transfer Learning”
  - Copy the top portion of a neural network trained on a large number of images
  - Only train the bottom portion based on the output from the top portion
  - Combine the two to make predictions
- Logged In Desktop classifier ends up 97.89% accurate

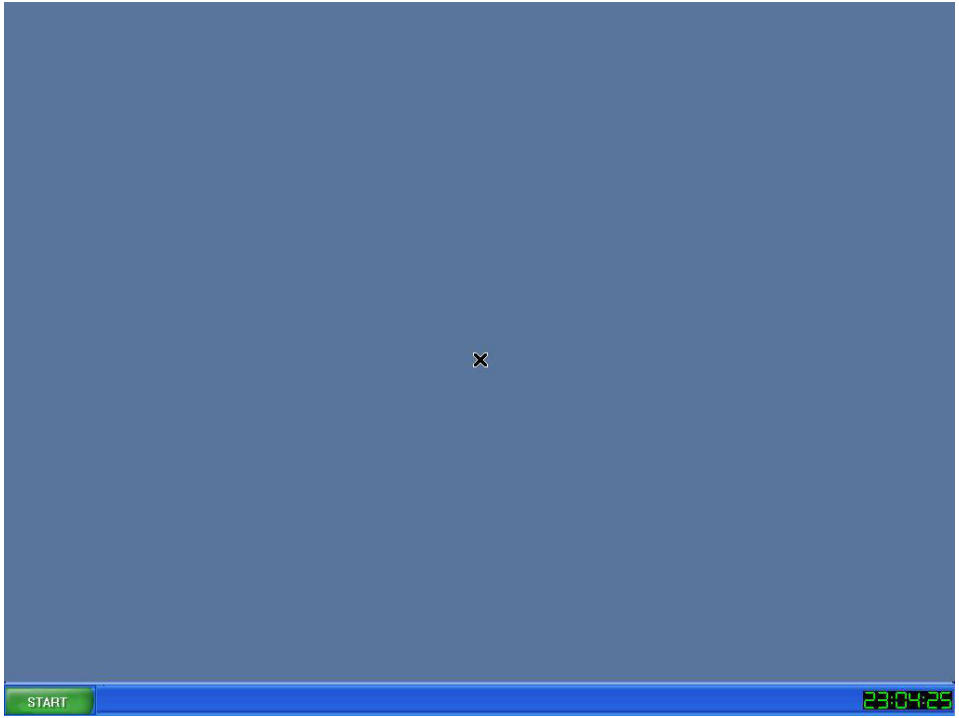
Logged In Desktop Classifier Findings



Logged In Desktop Classifier Findings



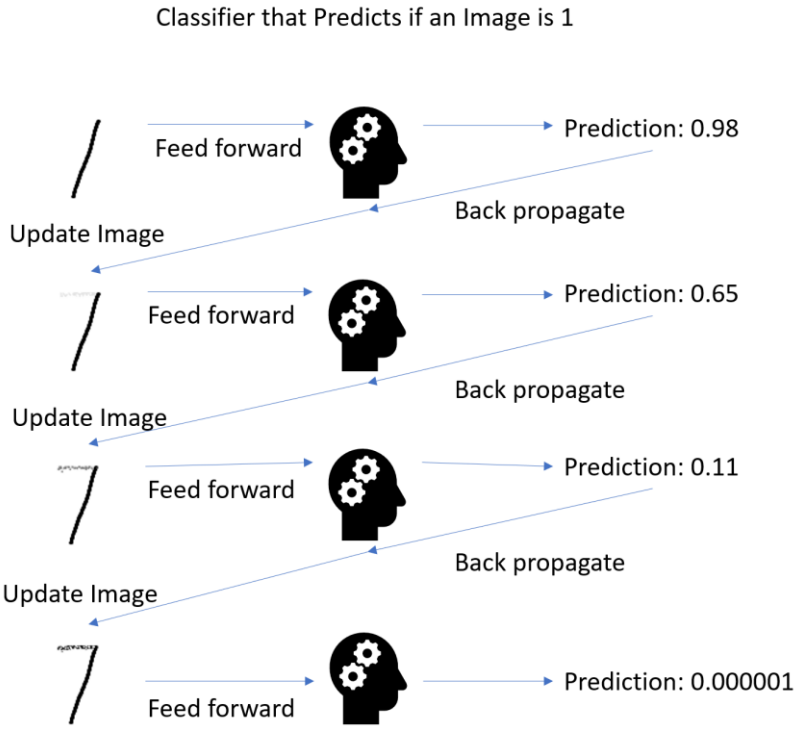
Logged In Desktop Classifier Findings



Logged In Desktop Classifier Findings

Generating Adversarial Samples

- Based on the same principals as training a neural network
- Feed in a sample to our model for classification
- Compare the output classification with the target classification we want
- Back propagate through the neural network (but not updating it) until we get to the sample
- Apply the remaining gradient to the image
- Continually repeat until the classification of the image flips



Adversarial Sample Generation Demo

- Running in a scientific computing environment called Jupyter Lab
  - A way to run Python (and other languages) experiments, and share results
  - Pytorch is the neural networking library being used primarily
- Start by loading the sample we want use as the starting point
- Load the model we want to attack (white box attack)
- Run an attack by applying changes to the whole image
- Run an attack by applying changes to a 100x100 square of the image
- Run an attack by imprinting a desired image and attacking the area around it

HACK #5
DEMO: ADVERSARIAL AI

(praise to the live demo gods!)

Hardening Models

- Making models more robust against attacks is referred to as “model hardening”
- There are many different strategies
- The suggested is simple; generate a lot of adversarial samples and add them to your training set with the original label
- After one round of this kind of adversarial training
  - Model accuracy went from 97.89% to 98.3%
  - Adversarial sample generation went from 0/530 failures to 100/530 failures
  - Mean distance from original image went from 3.11 to 4.03 (images had to change more to trick the model)

Adversarial Machine Learning And Defenses

- There is a continual back and forth between new defenses being created, and then new attacks being created to defeat those defenses
- Similar paradigm to the security industry with red vs blue team
- Implications of the security/robustness of a machine learning model could be quite significant given how much we rely on machine learning in everything from business operations to our day to day lives
- The best way to create secure/robust models is to regularly try to find ways to attack the models you create

Thank you!

- Questions and Answers?

@hackingexposed
Monthly HE Live sessions plus “versus” demos…