Why there's no such thing as cybersecurity

Matt Might, Ph.D.
School of Computing
University of Utah
matt.might.net
@mattmight

Why there's no such thing as cybersecurity yet

Matt Might, Ph.D.
School of Computing
University of Utah
matt.might.net
@mattmight

Fear

Hope

Why?

What?

How?

Why?

Crime


VISA
MasterCard

Schnucks breach will likely cost millions

HOW A PAYMENT GOES THROUGH THE SYSTEM

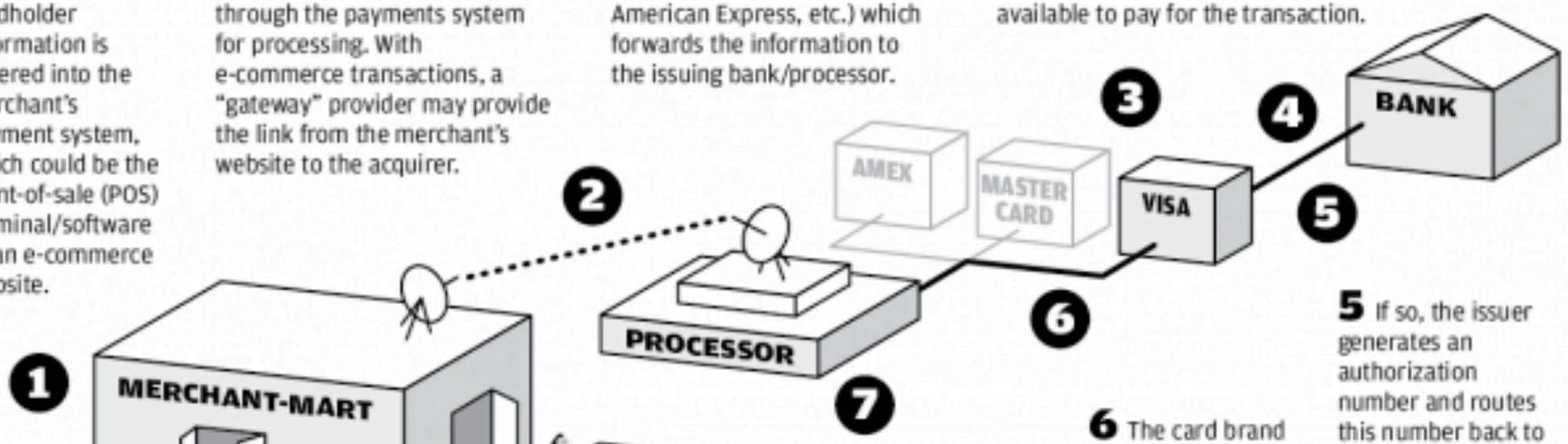
Information that moves through the payment system is encrypted for most of its travels, but at some points is decrypted so different parties can accept the data. That, experts say, is where data are vulnerable. Information is also at risk if stored without being encrypted or obfuscated.

1 The consumer selects a card for payment. The cardholder information is entered into the merchant's payment system, which could be the point-of-sale (POS) terminal/software or an e-commerce website.

2 The information is sent to an acquirer/payment processor, whose job it is to route the data through the payments system for processing. With e-commerce transactions, a "gateway" provider may provide the link from the merchant's website to the acquirer.

3 The acquirer/processor sends the data to the payment brand (e.g. Visa, MasterCard, American Express, etc.) which forwards the information to the issuing bank/processor.

4 The issuing bank/processor verifies that the card is legitimate, not reported lost or stolen, and that the account has the appropriate amount of credit/funds available to pay for the transaction.



5 If so, the issuer generates an authorization number and routes this number back to the card brand. With the authorization, the issuing bank agrees to fund the purchase on the consumer's behalf.

6 The card brand forwards the authorization code back to the acquirer/processor.

7 The acquirer/processor sends the authorization code back to the merchant.

8 The merchant concludes the sale with the customer.


Bank Name
Bank Name
Bank Name
Bank Name
Bank Name
Bank Name
1234 5678 9876 5432
1234
12/99
CARDHOLDER


Bank Name
Bank Name
1234 5678 9876 5432
1234
12/99
CARDHOLDER


Bank Name
Bank Name
1234 5678 9876 5432
1234
12/99
CARDHOLDER


Bank Name
Bank Name
1234 5678 9876 5432
1234
12/99
CARDHOLDER


Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER


Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER

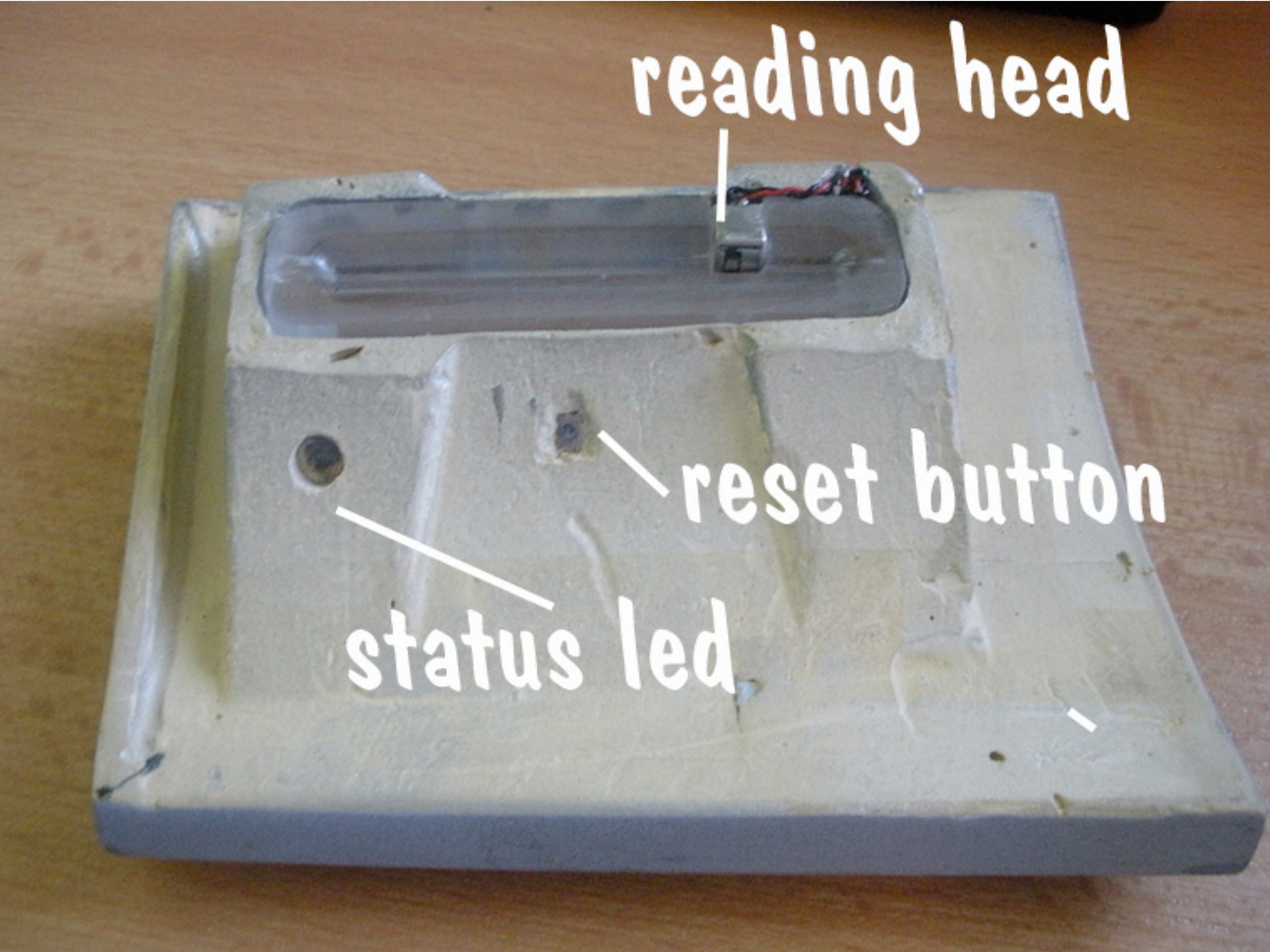

Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER


Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER


Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER


Bank Name
1234 5678 9876 5432
1234
VALID THRU
MONTH/YEAR
12/99
CARDHOLDER


RECEIPT
CARD
CASH


reading head
reset button
status led

SOCIAL SECURITY

XXX - XX - XXXX

THIS NUMBER HAS BEEN ESTABLISHED FOR

INSERT ID THEFT VICTIMS NAME HERE

SIGNATURE

South Carolina breach exposes 3.6M SSNs

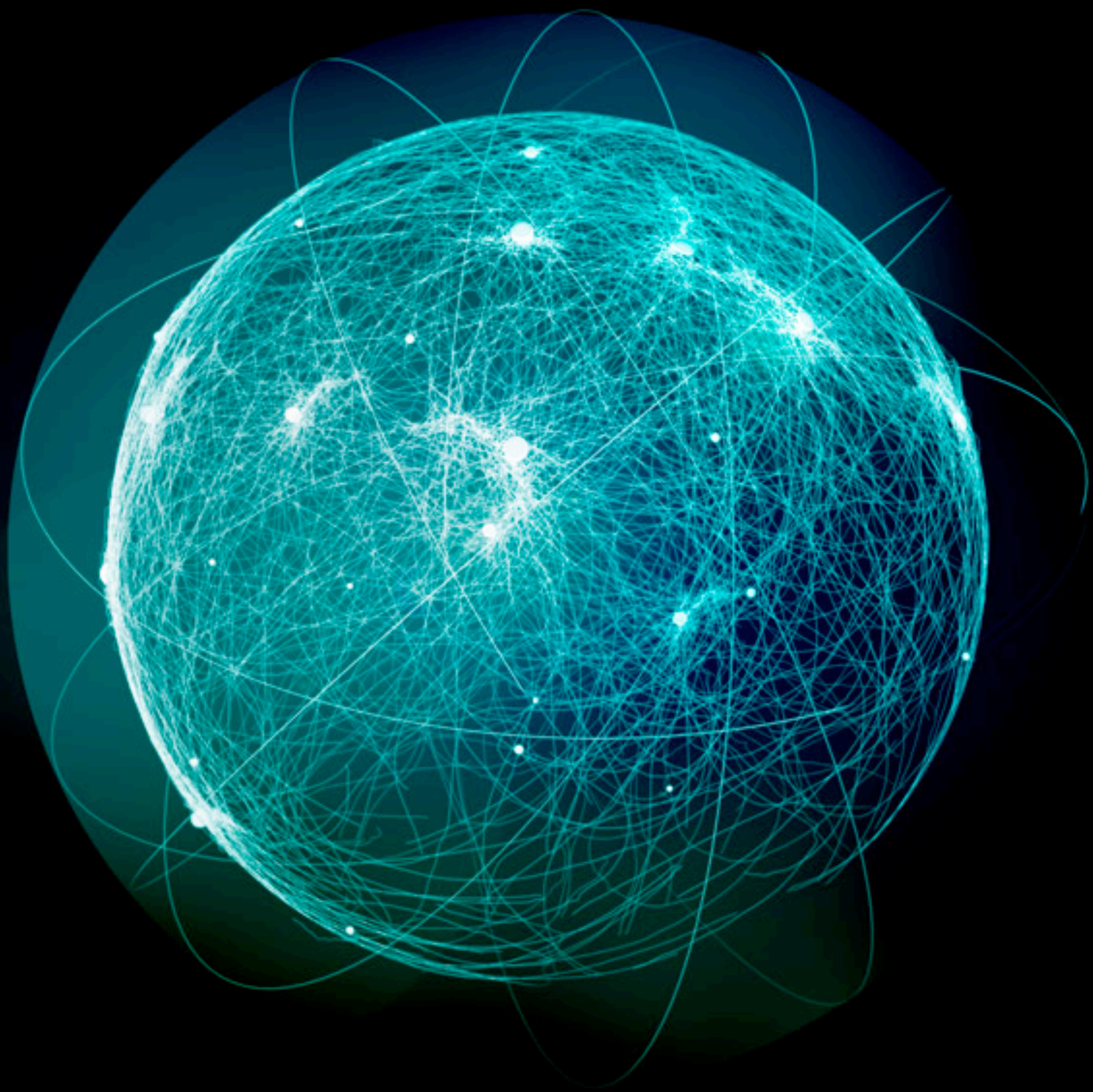
Another 387,000 credit and debit cards also exposed in Department of Revenue intrusion, but most were encrypted

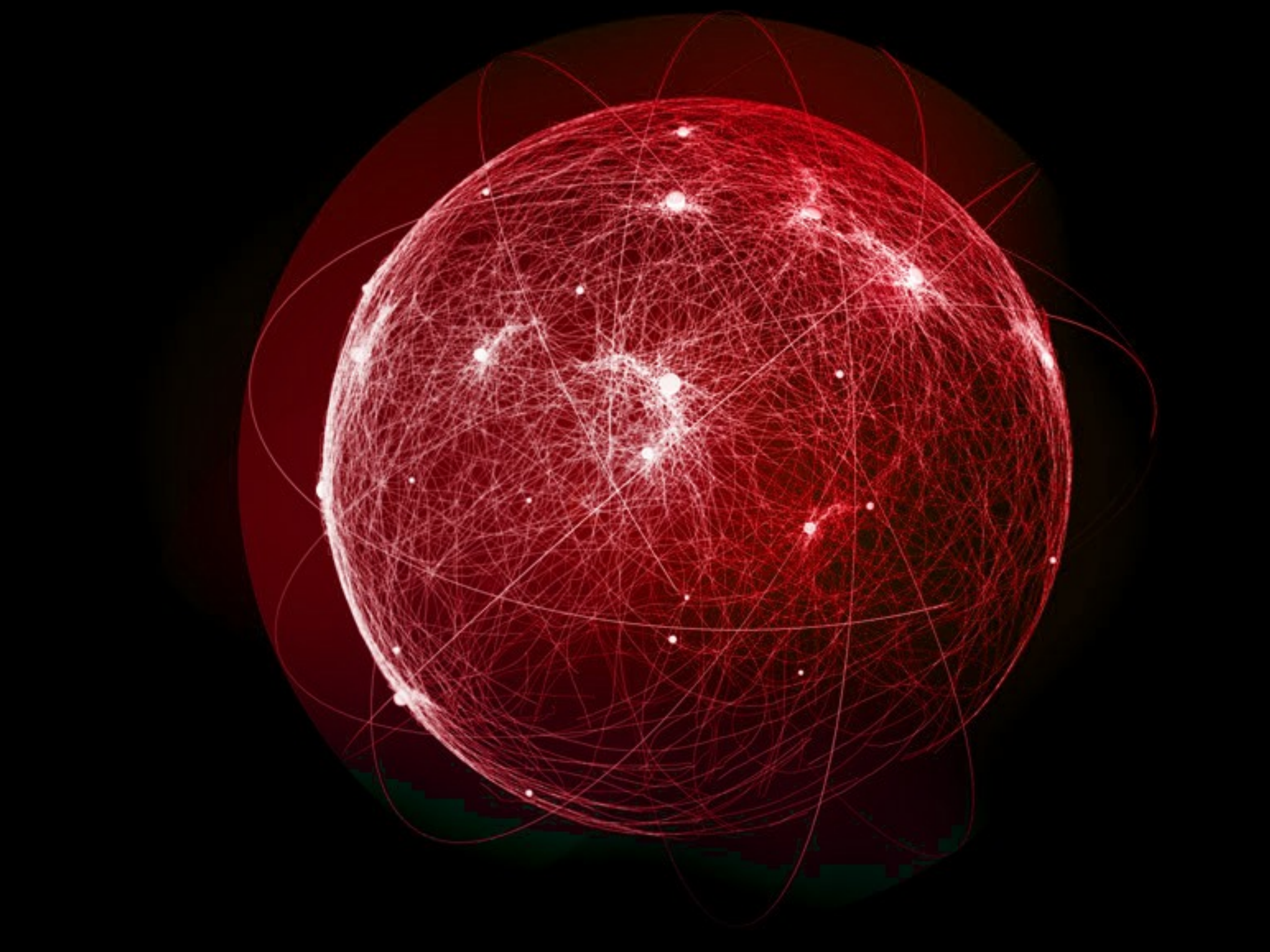
By Jaikumar Vijayan

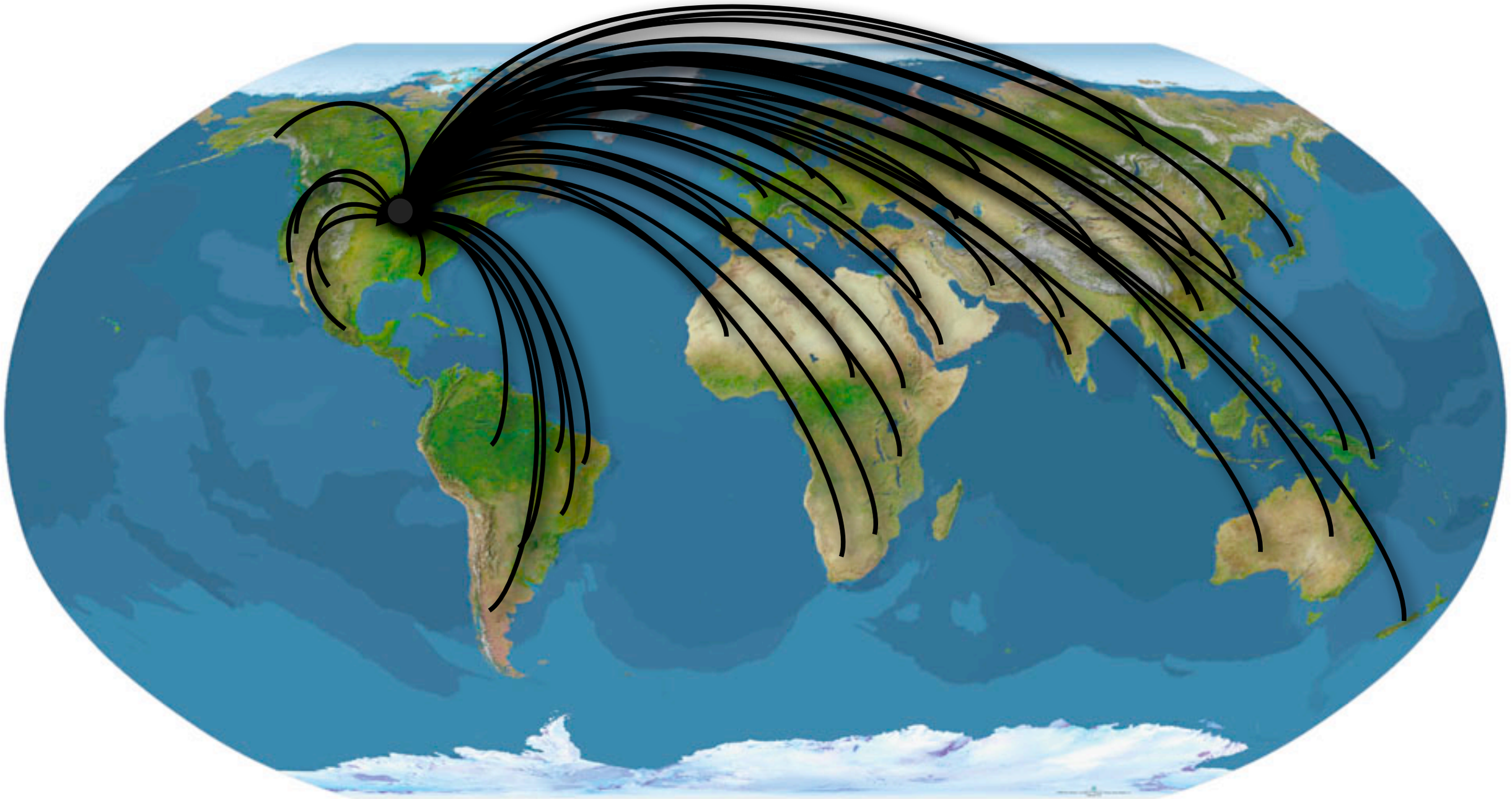
October 26, 2012 07:41 PM ET 7 Comments

Computerworld - In the biggest data compromise of the year, Social Security Numbers (SSN) belonging to about 3.6 million residents in South Carolina have been exposed in an intrusion into a computer at the state's Department of Revenue.

Another 387,000 credit and debit card numbers were also exposed in the September attack, the state Department of Revenue said in a statement Friday. However, out of that number only about 16,000 of the credit and debit cards were unencrypted, the department added. The SSNs, meanwhile, do not appear to have been encrypted.








SPAM®
CHOPPED PORK AND HAM
Serving Suggestion
Hormel®

War

საქართველოს ბანკი
BANK OF GEORGIA


NAT·
MDCCC
XXXIII
O·B·
MDCCC
XCVI
ALFR·
NOBEL


KISA


JICO


ZULU TIME
INDOC EAST
INDOC WEST
AFNOC Mission
Execute AFNETOPS/CC authority to command and control the operation and defense of the AF Network
AFNETOPS/CC Vision
Stripes Crew Assignments
CC – Lt McGhee
CC – TSgt Woolingham
AFNetOps C2 Relationship


NY
AF
85 5000

مرگ بر آمر

RQ170 – هواپیمای پیشرفته جاسوسی آمریکا

تسلط مدافعان ایران اسلامی بر تحرکات شیطان بزرگ

Google 搜索 手气不错

高级搜索
使用偏好
语言工具

所有网页 中文网页 简体中文网页 中国的网页

查询地址、搜索周边和规划路线

视频 图片 资讯 地图 博客搜索 热榜 网站导航

"All the News That's Fit to Print"

The New York Times

Late Edition

Today, strong wind, turning cooler, partly sunny, high 46. Tonight, partly cloudy, colder, low 28. Tomorrow, clouds and sun, a snow shower, high 34. Weather map, Page A24.

VOL. CLXII . . No. 56,033 © 2013 The New York Times NEW YORK, THURSDAY, JANUARY 31, 2013 $2.50

Hackers in China Attacked The Times for Last 4 Months

Computer Assaults Tied to Reporting on Premier

By NICOLE PERLROTH

SAN FRANCISCO — For the last four months, Chinese hackers have persistently attacked The New York Times, infiltrating its computer systems and getting passwords for its reporters and other employees.

After surreptitiously tracking the intruders to study their movements and help erect better defenses to block them, The Times and computer security experts, and Jim Yardley, The Times's South Asia bureau chief in India, who previously worked as bureau chief in Beijing.

"Computer security experts found no evidence that sensitive e-mails or files from the reporting of our articles about the Wen family were accessed, downloaded or copied," said Jill Abramson, executive editor of The Times.

The hackers tried to cloak the

ISRAELI AIRSTRIKE IN SYRIA TARGETS CONVOY, U.S. SAYS

LINK TO HEZBOLLAH SEEN

Arms Reported on Way to Lebanon — Denial by Damascus

By ISABEL KERSHNER


2001
B


RSA
SecurID®
159 759.
Secured by RSA

Hacktivism


SALT LAKE CITY
POLICE

ACS:Law

MasterCard

ELECTRONIC USE ONLY

VISA*

How?

TheXploit | Security Blog

HOME

Just Arrived: Malware Analyst's Cookbook

DEC 1ST Posted by Dustin in Exploit Development No comments

Author Michael Ligh was *very* gracious to send me a review copy of his new book Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code. I took a quick browse through it when I opened it and it looks *REALLY GOOD*. If it's anything like the articles on Michael's website, I know I'm in for a damn good read!

I'm planning on starting it this Saturday due to some other priorities so heads up for a review post in the future or check it out for yourself

Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code (Paperback)

By (author) Michael Ligh, Steven Adair, Blake Hartstein, Matthew Richard

List Price: $59.99 USD

CALENDAR

December 2010

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

« Nov

SEARCH


ABC
DEF
GHI
JKL
MNO
PRS
TUV
WXY
Spark Fun

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
504	152.15829	192.168.12.21	66.187.224.210	DNS	Standard query A www.redhat.com
505	152.24944	66.187.224.210	192.168.12.21	DNS	Standard query response A 209.132.177.50
506	152.25091	192.168.12.21	209.132.177.50	TCP	48890 > http [SYN] Seq=0 Len=0 MSS=1460 TSV=1535
507	152.31125	209.132.177.50	192.168.12.21	TCP	http > 48890 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len
508	152.31132	192.168.12.21	209.132.177.50	TCP	48890 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 TS
509	152.31154	192.168.12.21	209.132.177.50	HTTP	GET / HTTP/1.1
510	152.38737	209.132.177.50	192.168.12.21	TCP	http > 48890 [ACK] Seq=1 Ack=498 Win=6864 Len=0
511	152.40516	209.132.177.50	192.168.12.21	TCP	[TCP segment of a reassembled PDU]
512	152.40520	192.168.12.21	209.132.177.50	TCP	48890 > http [ACK] Seq=498 Ack=1369 Win=8576 Len
513	152.41351	209.132.177.50	192.168.12.21	TCP	[TCP segment of a reassembled PDU]
514	152.41356	192.168.12.21	209.132.177.50	TCP	48890 > http [ACK] Seq=498 Ack=2737 Win=11312 Le
515	152.45058	192.168.12.21	209.132.177.50	TCP	48891 > http [SYN] Seq=0 Len=0 MSS=1460 TSV=1535
516	152.47685	209.132.177.50	192.168.12.21	TCP	[TCP segment of a reassembled PDU]
517	152.47690	192.168.12.21	209.132.177.50	TCP	48890 > http [ACK] Seq=498 Ack=4105 Win=14048 Le

```
▷ Frame 507 (74 bytes on wire, 74 bytes captured)
▷ Ethernet II, Src: Amit_04:ae:54 (00:50:18:04:ae:54), Dst: Intel_e3:01:f5 (00:0c:f1:e3:01:f5)
▷ Internet Protocol, Src: 209.132.177.50 (209.132.177.50), Dst: 192.168.12.21 (192.168.12.21)
▽ Transmission Control Protocol, Src Port: http (80), Dst Port: 48890 (48890), Seq: 0, Ack: 1, Len: 0
    Source port: http (80)
    Destination port: 48890 (48890)
    Sequence number: 0    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 40 bytes
  ▷ Flags: 0x12 (SYN, ACK)
    Window size: 5792
    Checksum: 0x99db [correct]
  ▷ Options: (20 bytes)
  ▷ [SEQ/ACK analysis]
```

```
0000  00 0c f1 e3 01 f5 00 50  18 04 ae 54 08 00 45 00   .......P ...T..E.
0010  00 3c 00 00 40 00 35 06  f6 47 d1 84 b1 32 c0 a8   .<..@.5. .G...2..
0020  0c 15 00 50 be fa b5 36  ce 18 e0 bb b5 58 a0 12   ...P...6 .....X..
0030  16 a0 99 db 00 00 02 04  05 64 04 02 08 0a 10 1d   ........ .d......
0040  ee de 5b 81 15 29 01 03  03 02                     ..[..).. ..
```

Source Port (tcp.srcport), 2 P: 1096 D: 1096 M: 0 Drops: 0

Mozilla Firefox

(Untitled)

Google

Firesheep

Stop Capturing

eric+google@codebutler.com
Google

Ian Gallagher
Facebook

neg9
Twitter

cdine
Flickr

facebook

Search

Ian Gallagher
Edit My Profile

News Feed
Messages
Events 1
Friends

Create Group...

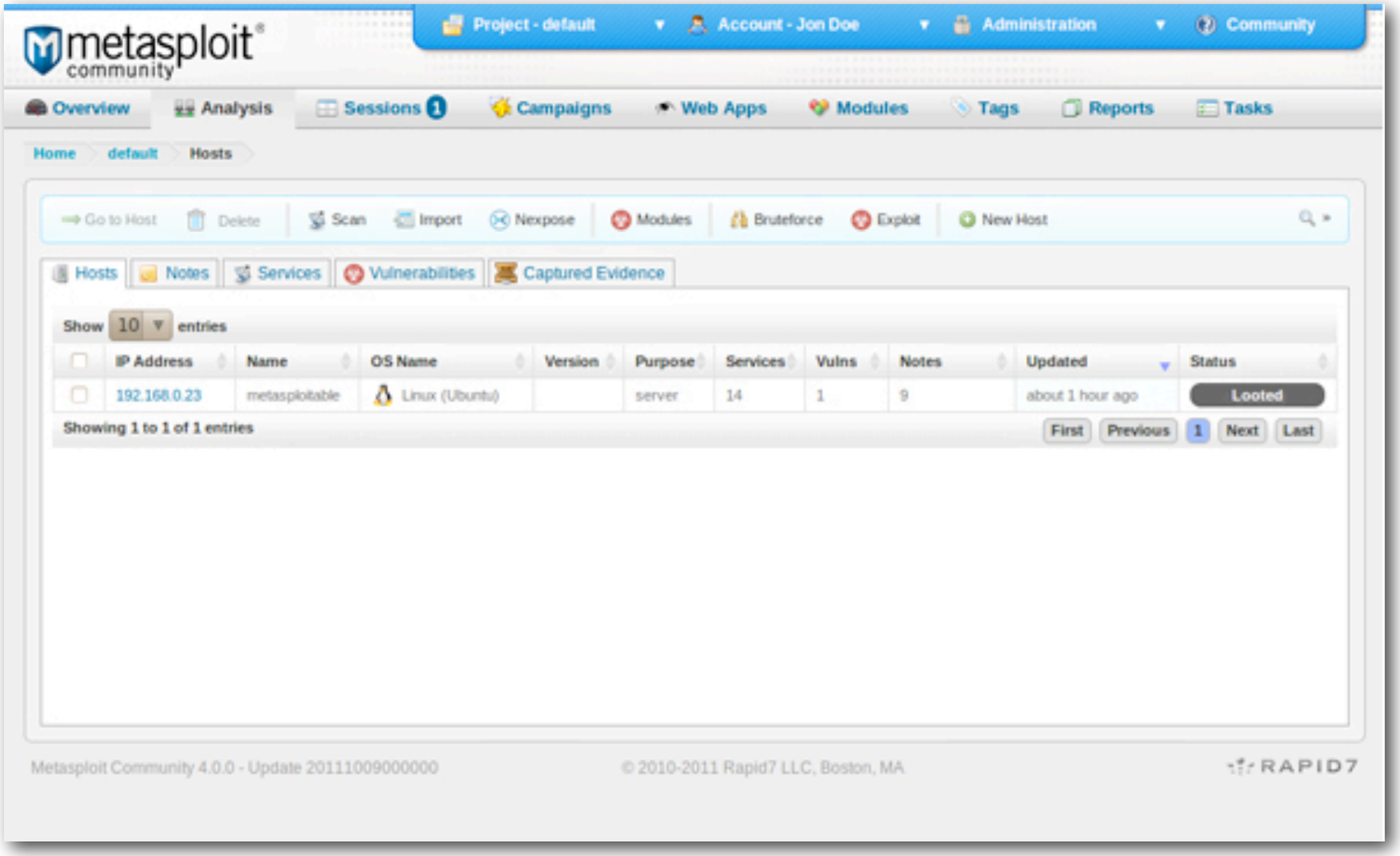
News Feed

What's on your mind?

Ashley Winter
realized i really
for some fake r

indeed.

FoxyProxy: Disabled


metasploit community
Project - default
Account - Jon Doe
Administration
Community
Overview
Analysis
Sessions 1
Campaigns
Web Apps
Modules
Tags
Reports
Tasks
Home
default
Hosts
Go to Host
Delete
Scan
Import
Nexpose
Modules
Bruteforce
Exploit
New Host
Hosts
Notes
Services
Vulnerabilities
Captured Evidence
Show 10 entries
IP Address
Name
OS Name
Version
Purpose
Services
Vulns
Notes
Updated
Status
192.168.0.23
metasploitable
Linux (Ubuntu)
server
14
1
9
about 1 hour ago
Looted
Showing 1 to 1 of 1 entries
First
Previous
1
Next
Last
Metasploit Community 4.0.0 - Update 20111009000000
© 2010-2011 Rapid7 LLC, Boston, MA
RAPID7

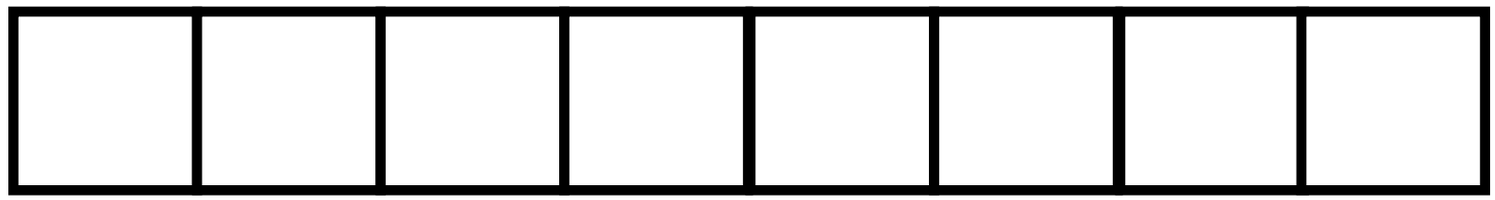
Why so vulnerable?

Programmers screw up!

vulnerability = mistake

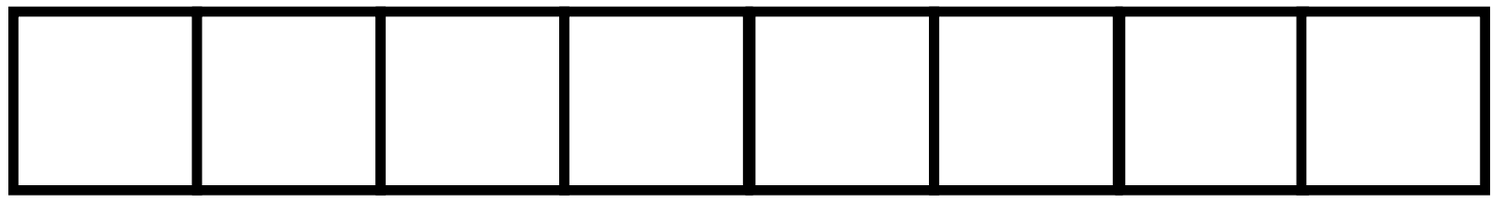
no mistake = no vulnerability

Example: Buffer overflow



John Day

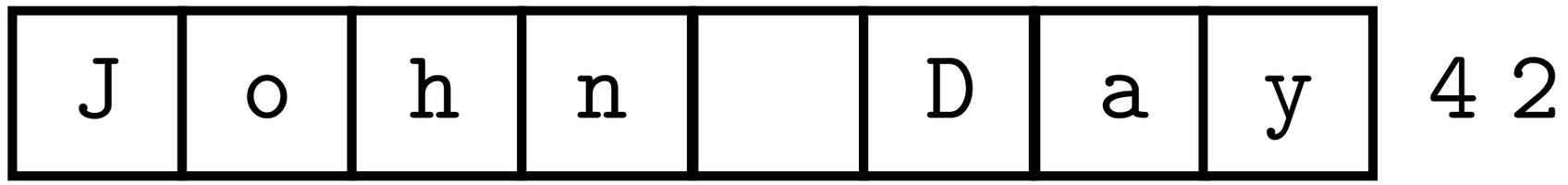
J	o	h	n		D	a	y

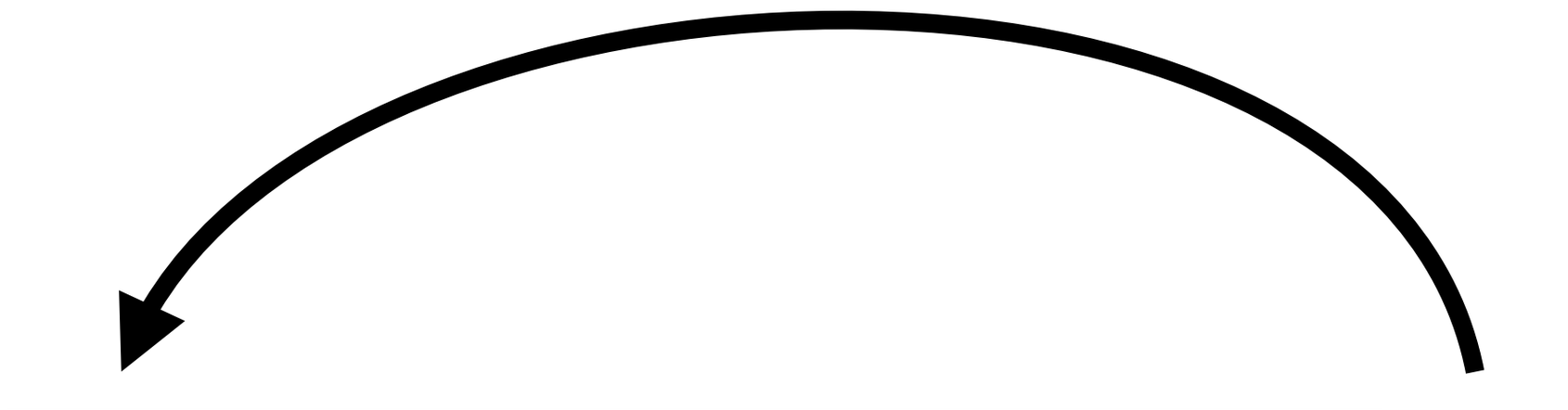


John Day$*

J	o	h	n		D	a	y

$ *


J o h n D a y
4 2


J o h n D a y
4 2

t	a	k	e	o	v	e	r

42

Defense

Discovery

Private disclosure

Exploit creation

Patch release → Patched


BANG!

Discovery

Patch release → Patched

Public disclosure → Exploit creation

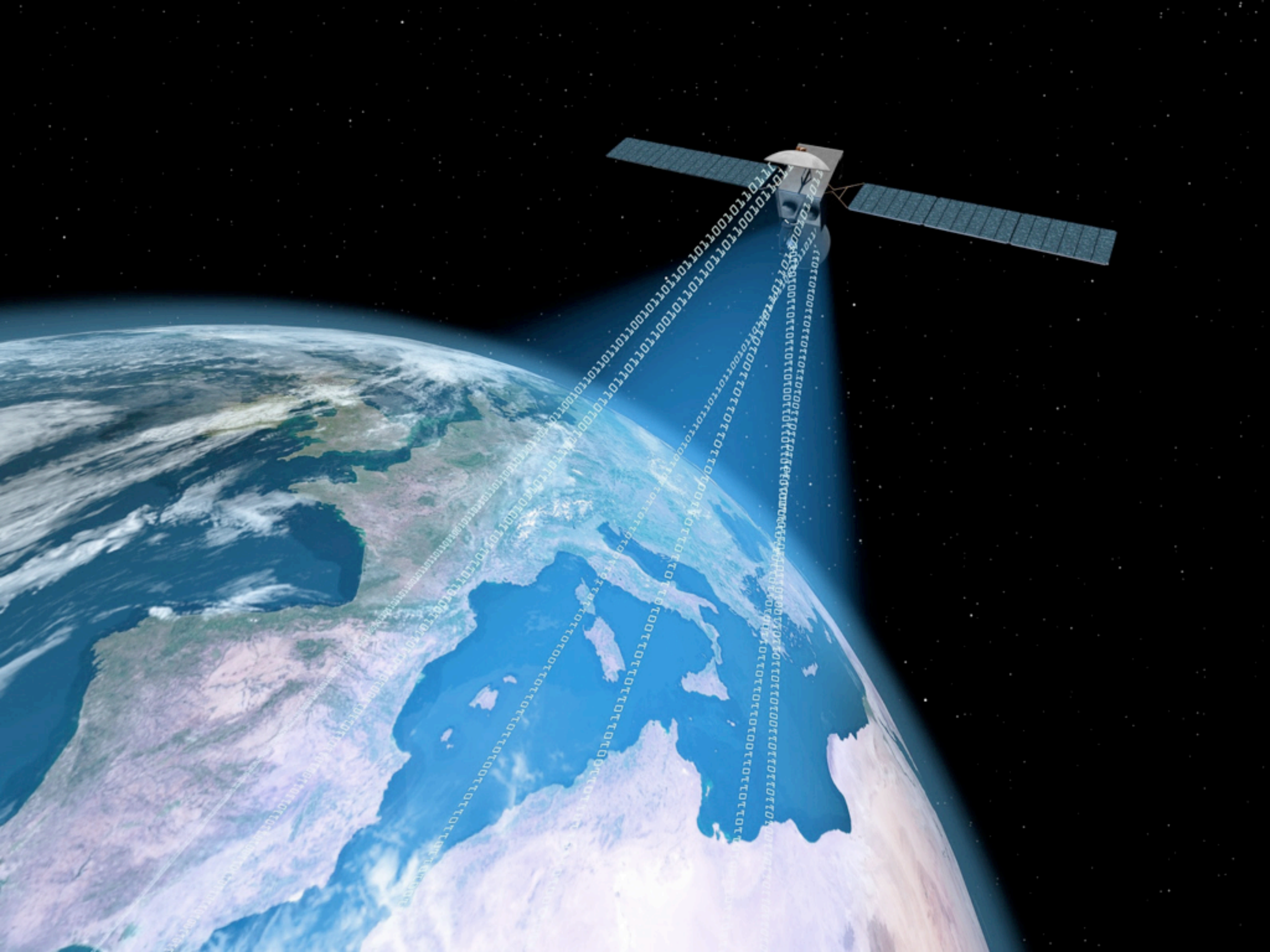
Discovery → Exploit creation → Attack

Discovery → Weapon creation

6 P's

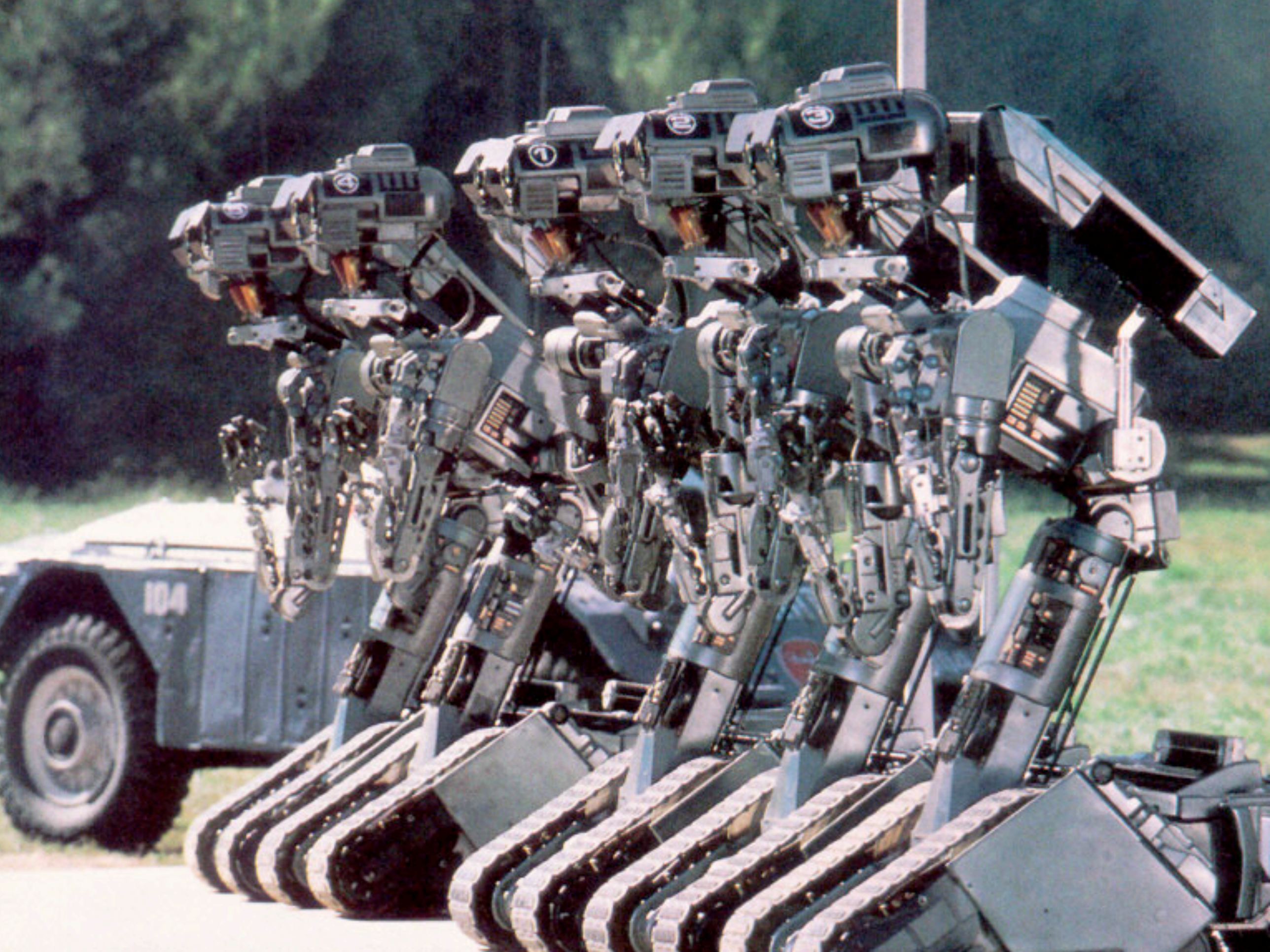
Passwords.
Patches.
Popularity.
Protection.
Preparation.
Prayer.

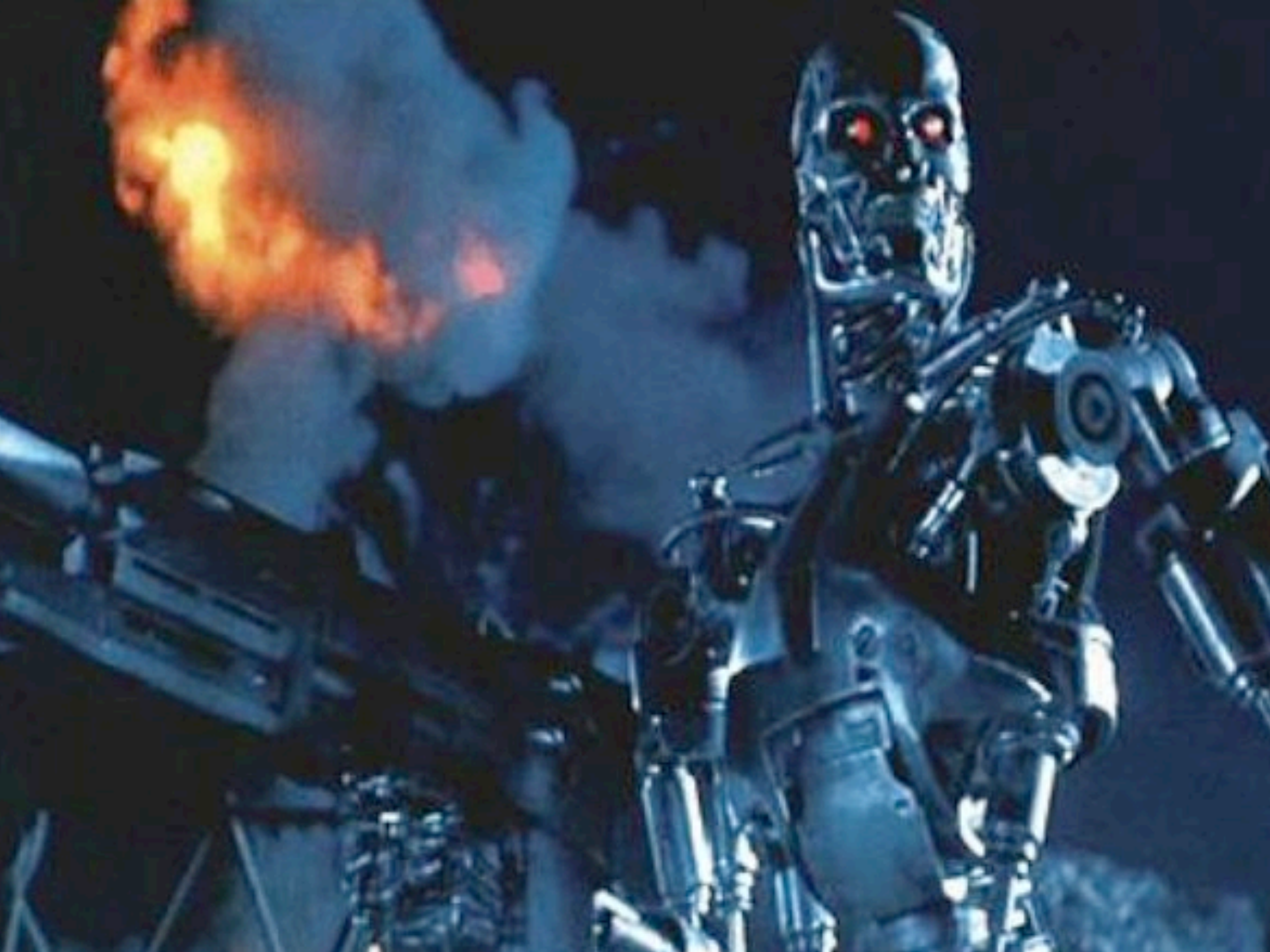
Future




NY
AF
85 5000


#3



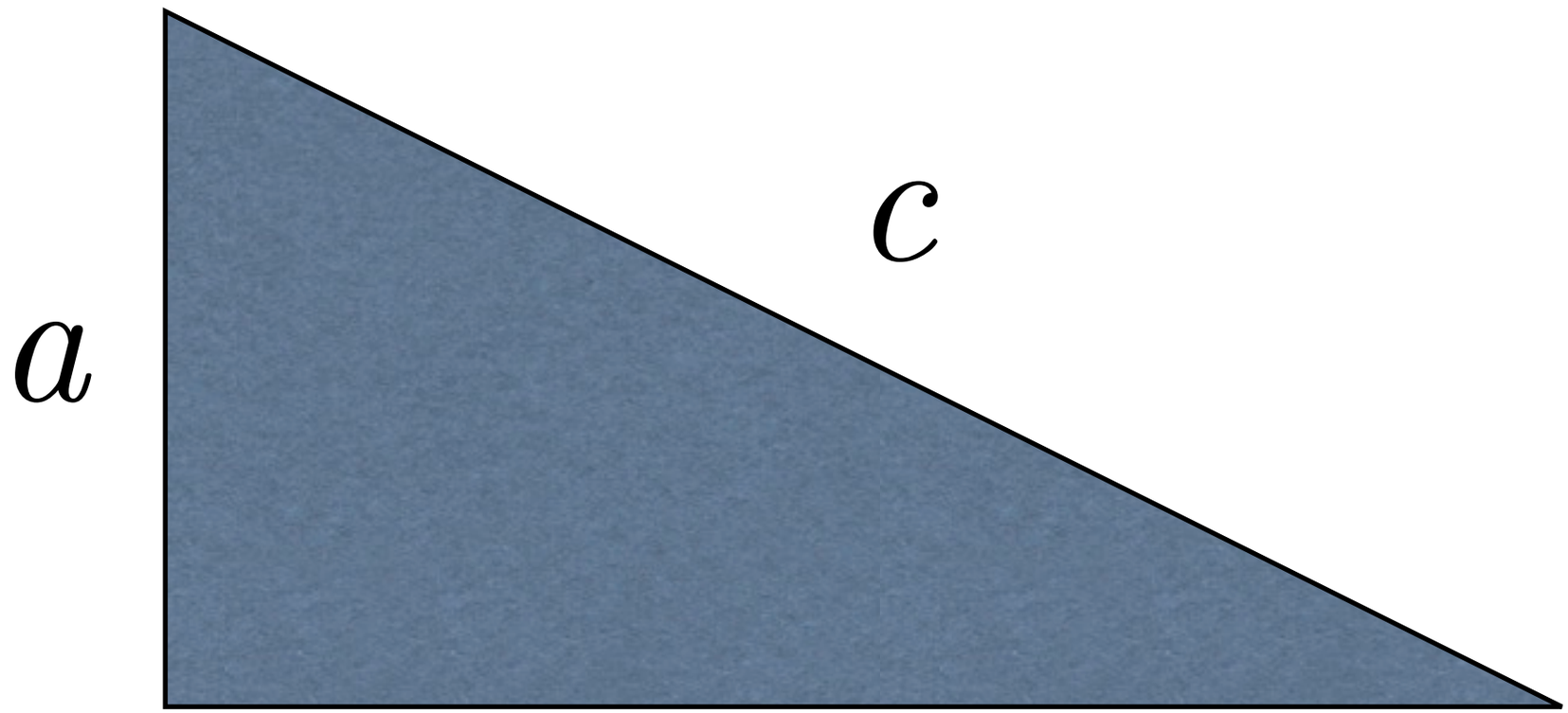


How?

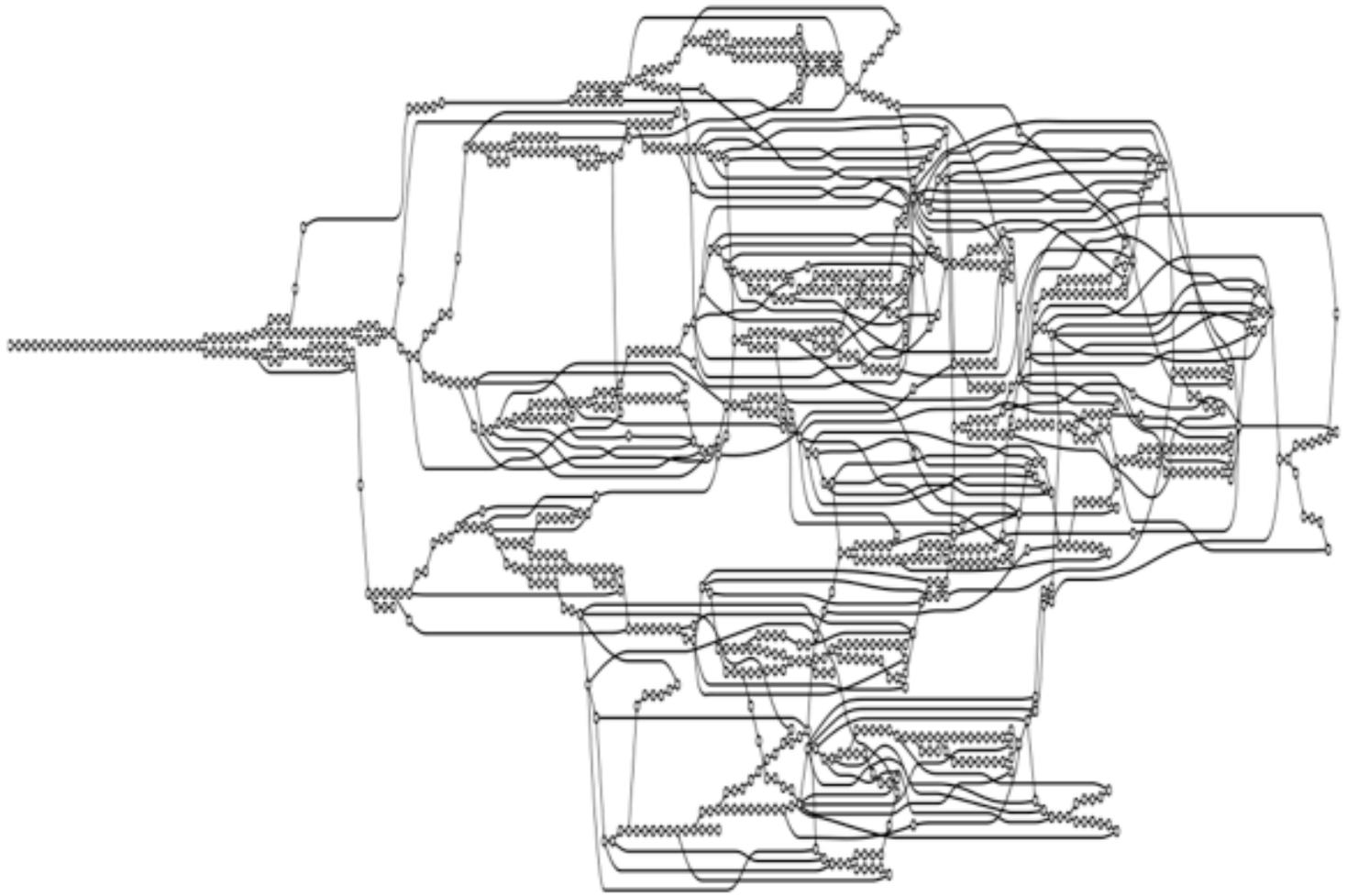
Math.

Lots of

Math.



$$a^2 + b^2 = c^2$$

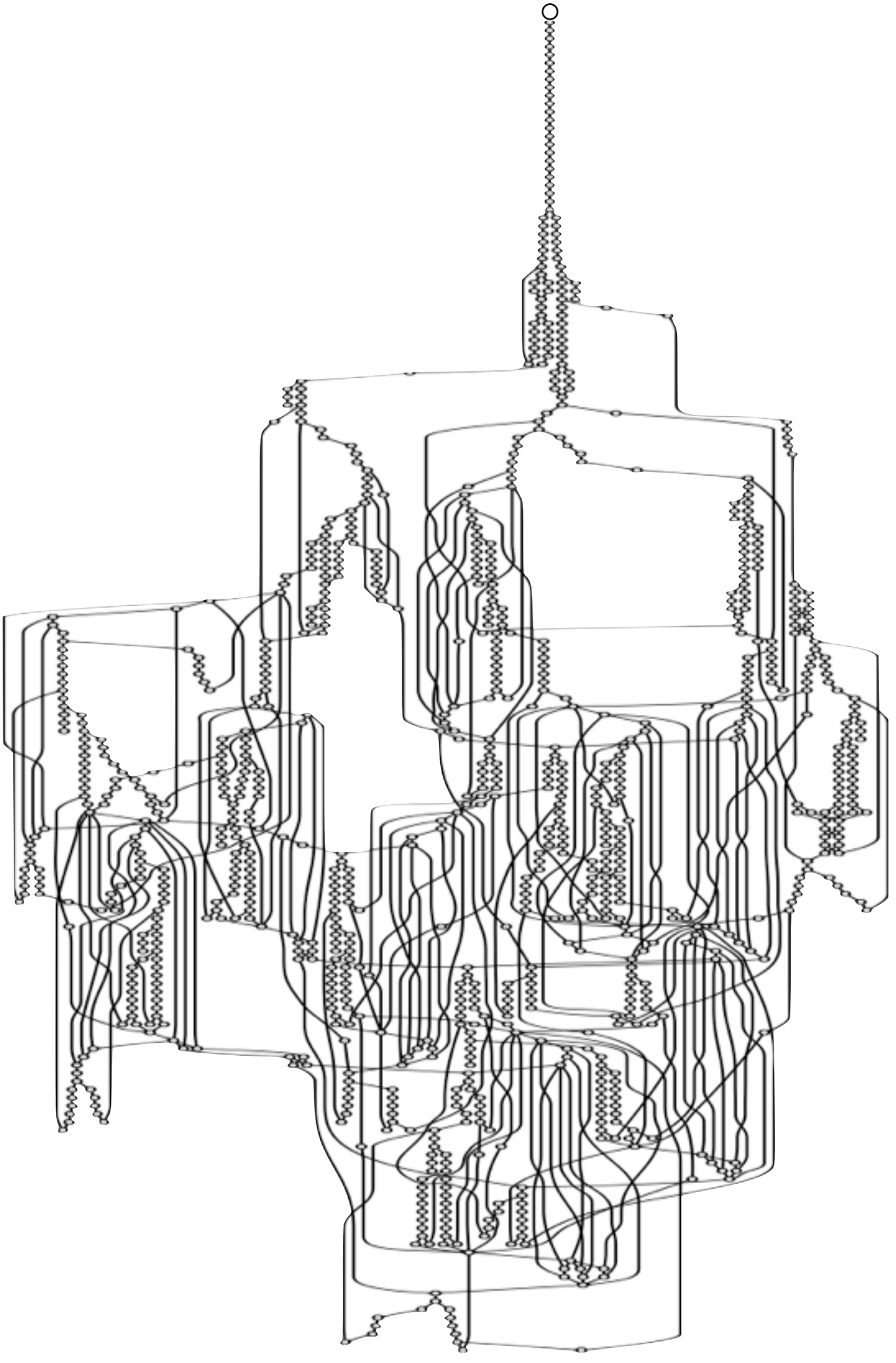


$$\forall \sigma \in \{\sigma' \mid \mathcal{I}(program) \Rightarrow^* \sigma'\} : \sigma \in \mathsf{WF}$$

How?

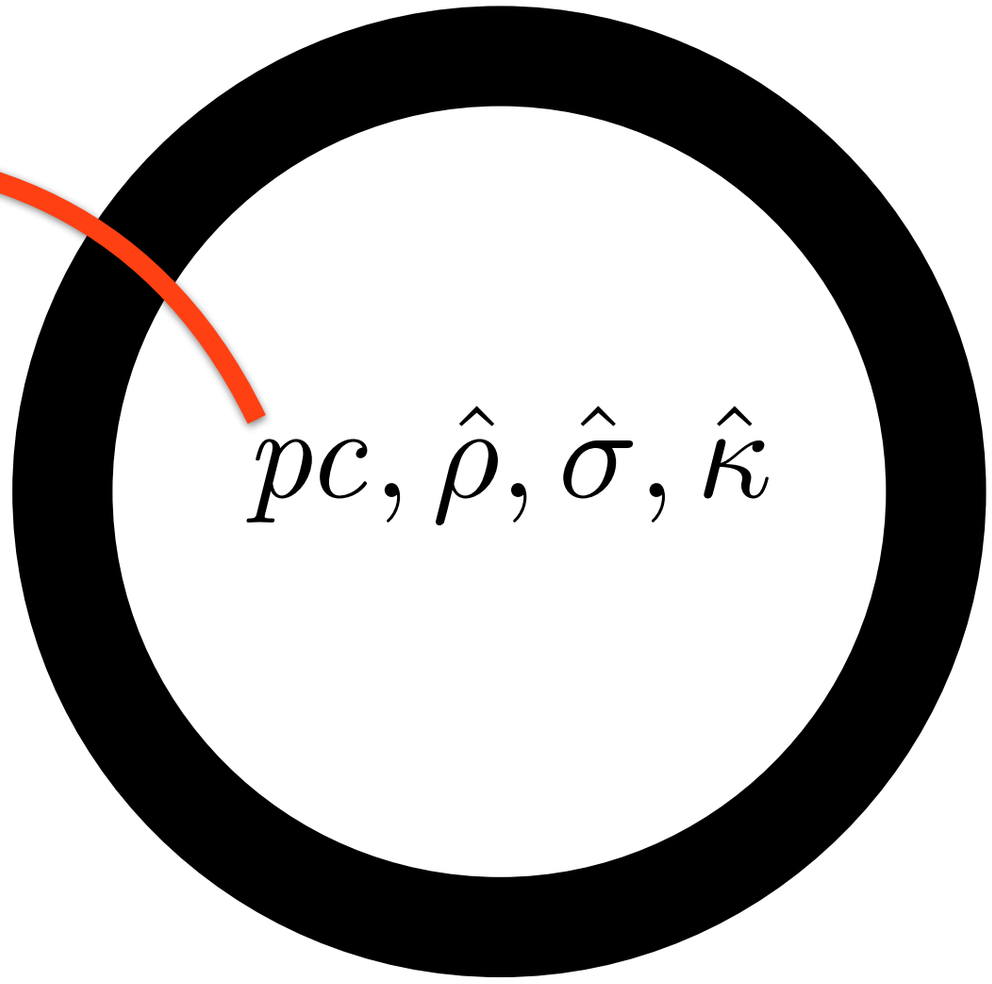
-

```
.class MyActivity

 .method public MyActivity
    invokedynamic activateMic
 .end method

.end class
```

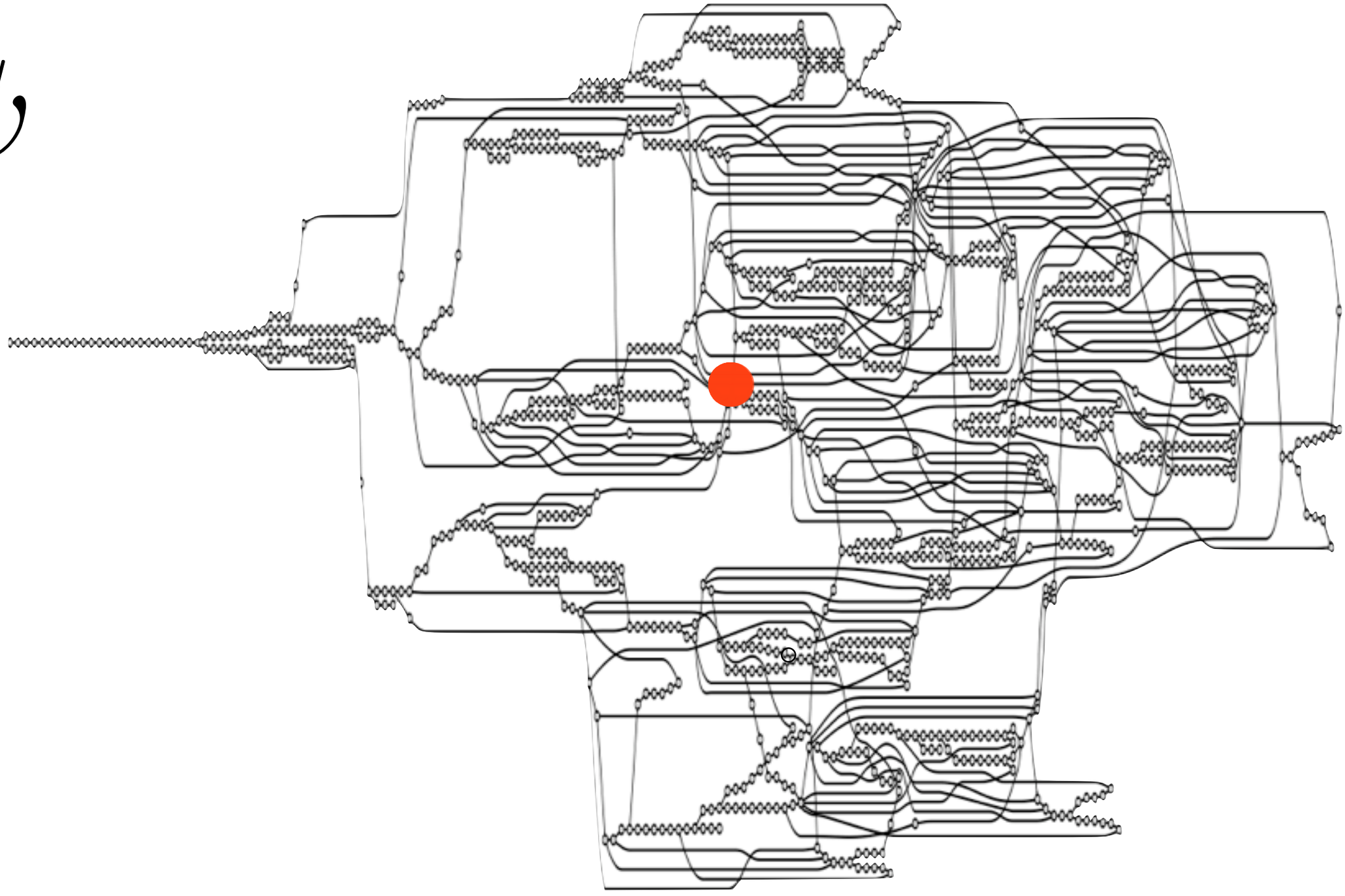


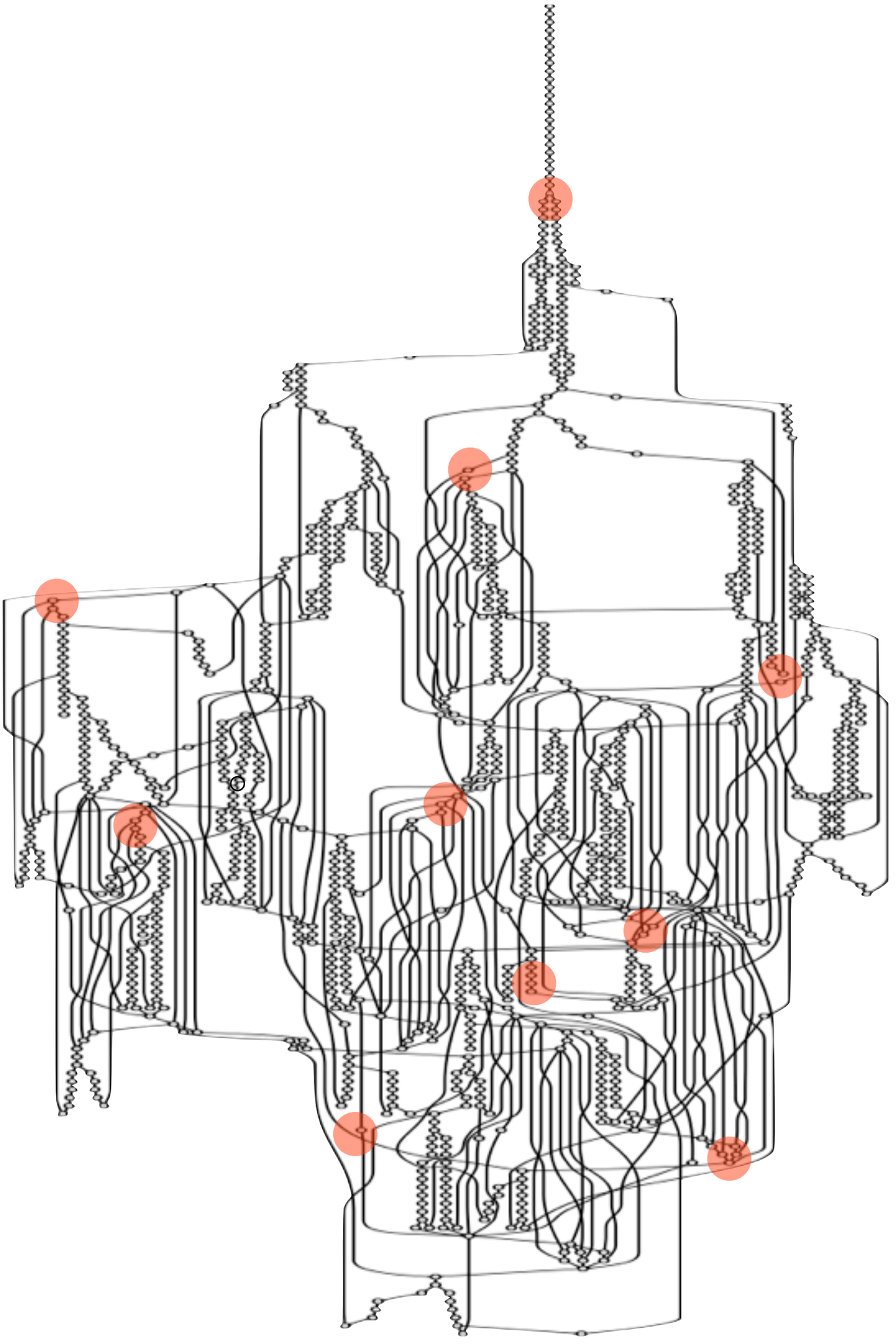
Vulnerability!

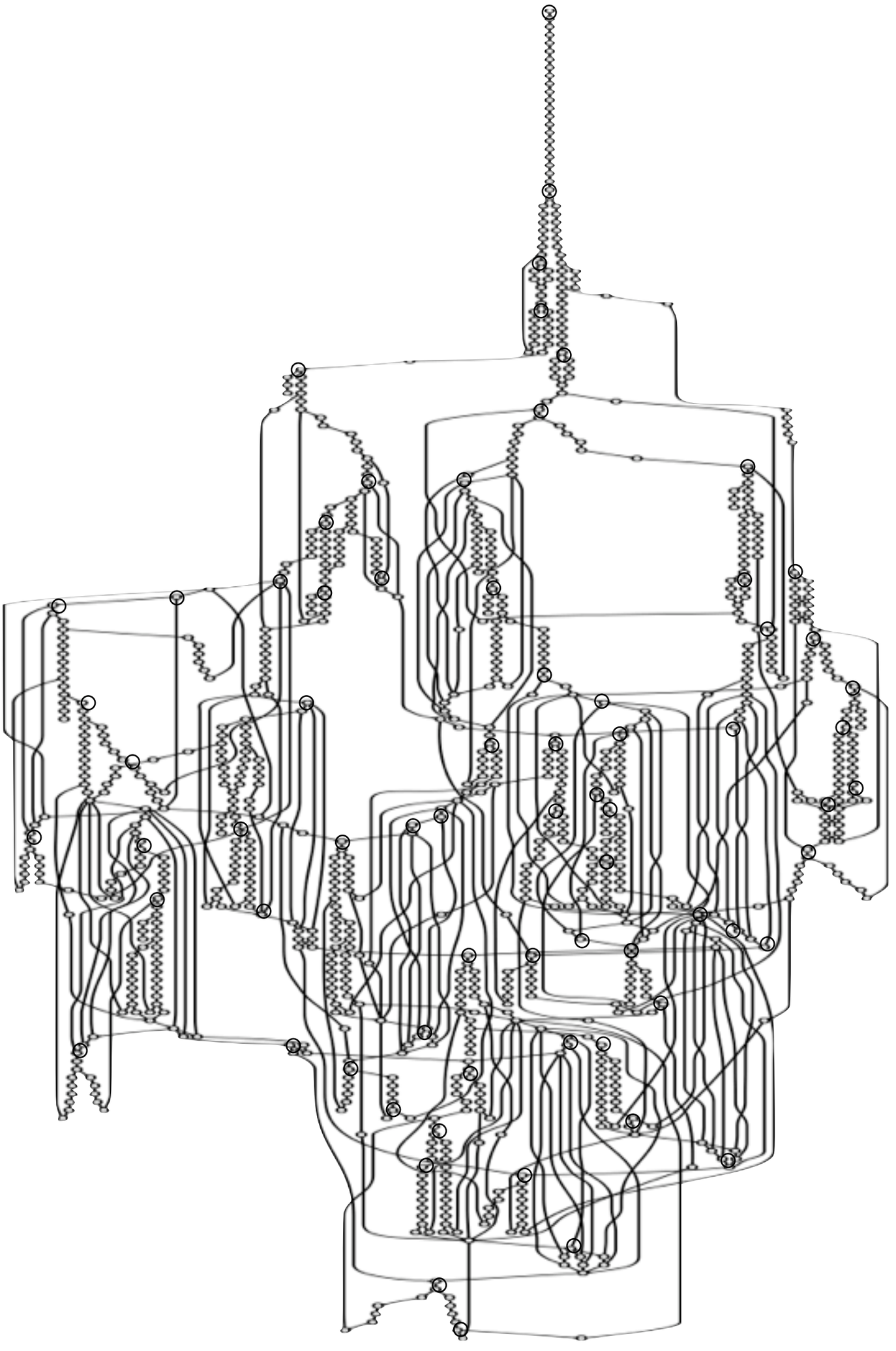
```
.class MyActivity

 .method public MyActivity
    invokedynamic activateMic
 .end method

.end class
```



ψ







GPS

Disk

Disk

Net

Contacts

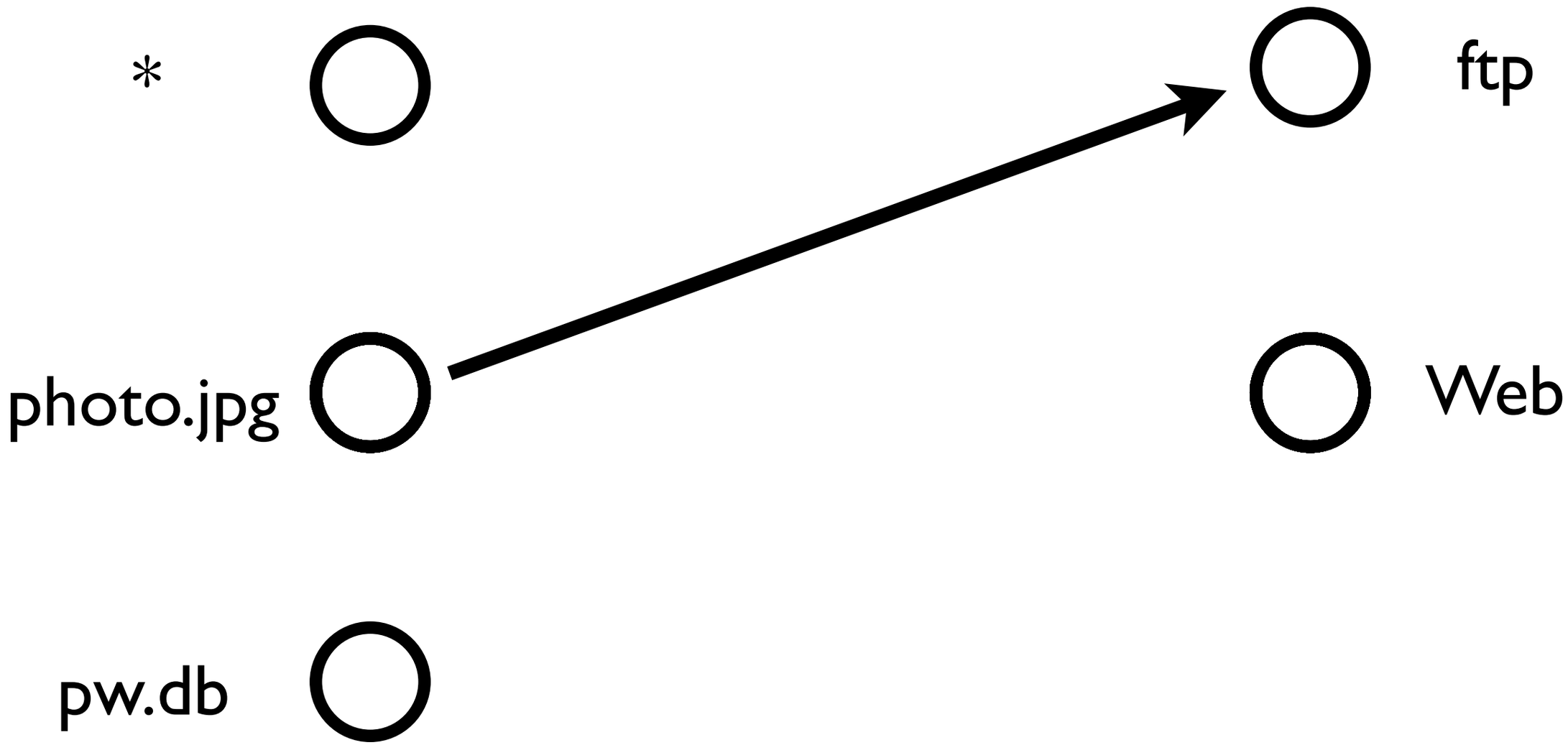
RPC

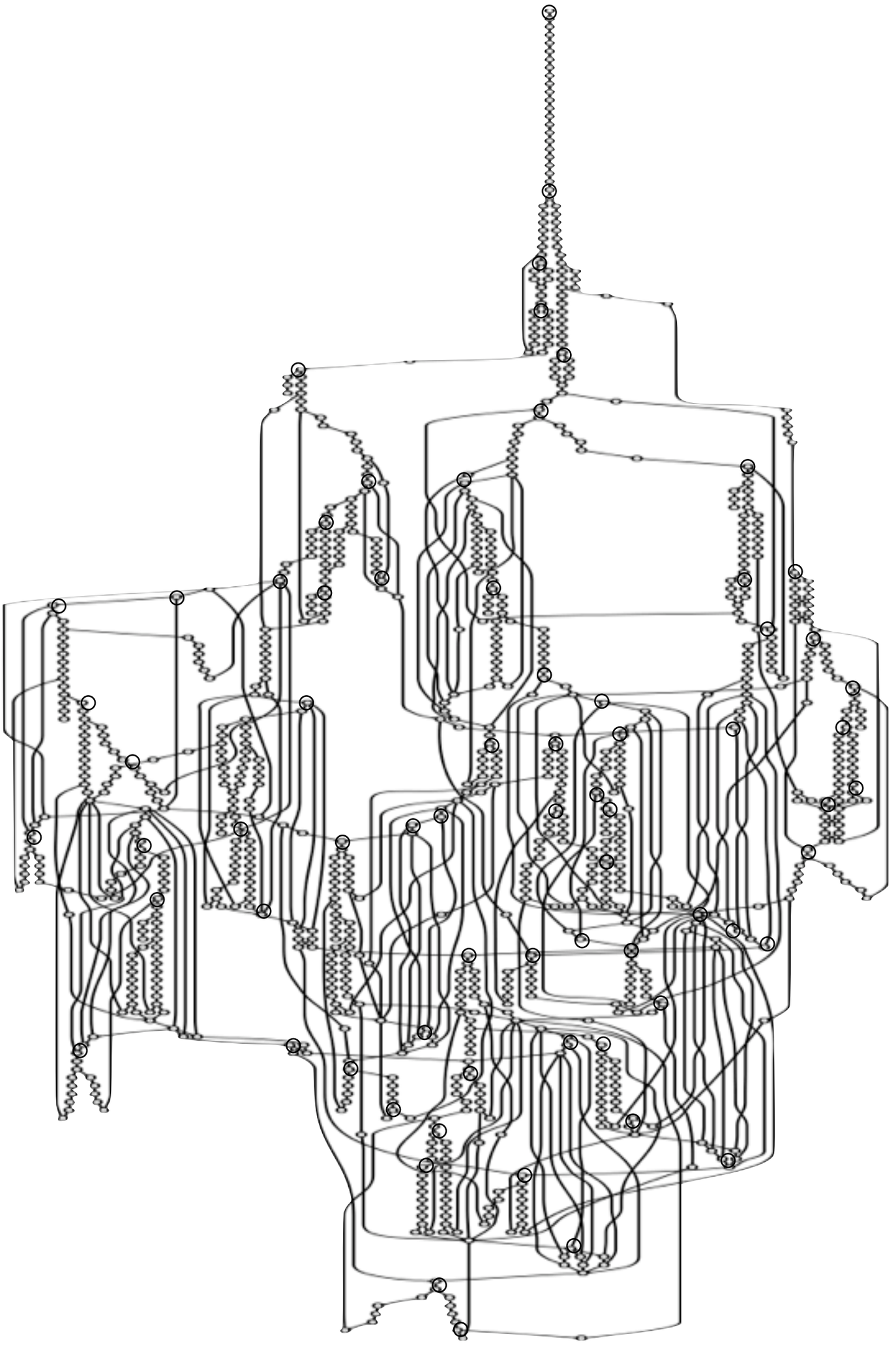
*

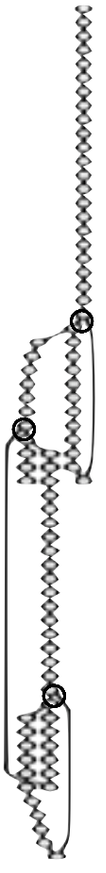
photo.jpg

Net

pw.db


*
ftp
photo.jpg
Web
pw.db





$$F = ma$$

$$\langle \texttt{nop} :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle$$

$$\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(r_s, fp)], \kappa\rangle$$

$$\langle \texttt{return-void} :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle$$

$$\langle \texttt{return-object}(r) :: \vec{stmt}', fp', \sigma, \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(\texttt{ret}, fp) \mapsto \sigma(n, fp')], \kappa\rangle$$

$$\langle \texttt{const}(r, c) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto c], \kappa\rangle$$

$$\langle \texttt{throw}^{\ell}(r) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell'), fp', \sigma[(\texttt{exn}, fp') \mapsto \sigma(r, fp)], \kappa'\rangle$$
$$\text{where } (\ell', fp', \kappa') = \mathcal{H}(\ell, fp, \kappa)$$

$$\langle \texttt{goto}(\ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle$$

$$\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r, fp) \mapsto o], \kappa\rangle$$
$$\text{where } o = new(\varsigma)$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{S}(\ell), fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) = \sigma(r', fp)$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma, \kappa\rangle \text{ if } \sigma(r, fp) \neq \sigma(r', fp)$$

$$\langle \texttt{iget}(r_d, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto \sigma(a)], \kappa\rangle$$
$$\text{where } \sigma(r_s, fp) = o \text{ and } o.field = a$$

$$\langle \texttt{iput}(r_v, r_s, field) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[a \mapsto \sigma(r_v, fp)], \kappa\rangle$$
$$\text{where } \sigma(r_s, fp) = o \text{ and } o.field = a$$

$$\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{M}(id), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle$$
$$\text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)]$$
$$fp' = alloc(\varsigma)$$

$$\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \mathcal{V}(id, \sigma(r_0, fp)), fp', \sigma', \mathbf{fnk}(\vec{stmt}, fp, \kappa)\rangle$$
$$\text{where } \sigma' = \sigma[(0, fp') \mapsto \sigma(r_0, fp), \ldots, (n, fp') \mapsto \sigma(r_n, fp)]$$
$$fp' = alloc(\varsigma)$$

$$\langle \mathit{unop}(r_d, r_s) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle$$
$$\text{where } v = \delta(\mathit{unop}, \sigma(r_s, fp))$$

$$\langle \mathit{binop}(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, fp, \sigma[(r_d, fp) \mapsto v], \kappa\rangle$$
$$\text{where } v = \delta(\mathit{binop}, \sigma(r_1, fp), \sigma(r_2, fp))$$

$$\langle \texttt{nop} :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle$$

$$\langle \texttt{move-object}(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(r_s, \hat{fp})], \hat{\kappa}, \hat{t}\rangle$$

$$\langle \texttt{return-void} :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa)$$

$$\langle \texttt{return-object}(r) :: \vec{stmt}', \hat{fp}', \hat{\sigma}, \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa)\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(\texttt{ret}, \hat{fp}) \mapsto \hat{\sigma}(n, \hat{fp}')], \hat{\kappa}\rangle \text{ if } \hat{\kappa} \in \hat{\sigma}(\hat{a}_\kappa)$$

$$\langle \texttt{const}(r, c) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto c], \hat{\kappa}, \hat{t}\rangle$$

$$\langle \texttt{throw}^\ell(r) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle \longmapsto \langle \mathcal{S}(\ell'), fp', \hat{\sigma} \sqcup [(\texttt{exn}, \hat{fp}') \mapsto \hat{\sigma}(r, \hat{fp})], \hat{\kappa}'\rangle$$
$$\text{where } (\ell', \hat{fp}', \hat{\kappa}') \in \widehat{\mathcal{H}}_{\hat{\sigma}}(\ell, \hat{fp}, \hat{\kappa})$$

$$\langle \texttt{goto}(\ell) :: \vec{stmt}', \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle$$

$$\langle \texttt{new-instance}(r, \tau) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r, \hat{fp}) \mapsto o], \hat{\kappa}, \hat{t}\rangle$$
$$\text{where } o = \widehat{new}(\varsigma)$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \mathcal{S}(\ell), \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle$$
$$\text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 = v_2$$

$$\langle \texttt{if-eq}(r, r', \ell) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle$$
$$\text{if } \exists v_1 \in \hat{\sigma}(r, \hat{fp}), \exists v_2 \in \hat{\sigma}(r', \hat{fp}).v_1 \neq v_2$$

$$\langle \texttt{iget}(r_d, r_s, \mathit{field}) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto \hat{\sigma}(a)], \hat{\kappa}, \hat{t}\rangle$$
$$\text{where } \hat{\sigma}(r_s, fp) \ni o \text{ and } o.\mathit{field} = a$$

$$\langle \texttt{iput}(r_v, r_s, \mathit{field}) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [a \mapsto \hat{\sigma}(r_v, \hat{fp})], \hat{\kappa}, \hat{t}\rangle$$
$$\text{where } \hat{\sigma}(r_s, \hat{fp}) \ni o \text{ and } o.\mathit{field} = a$$

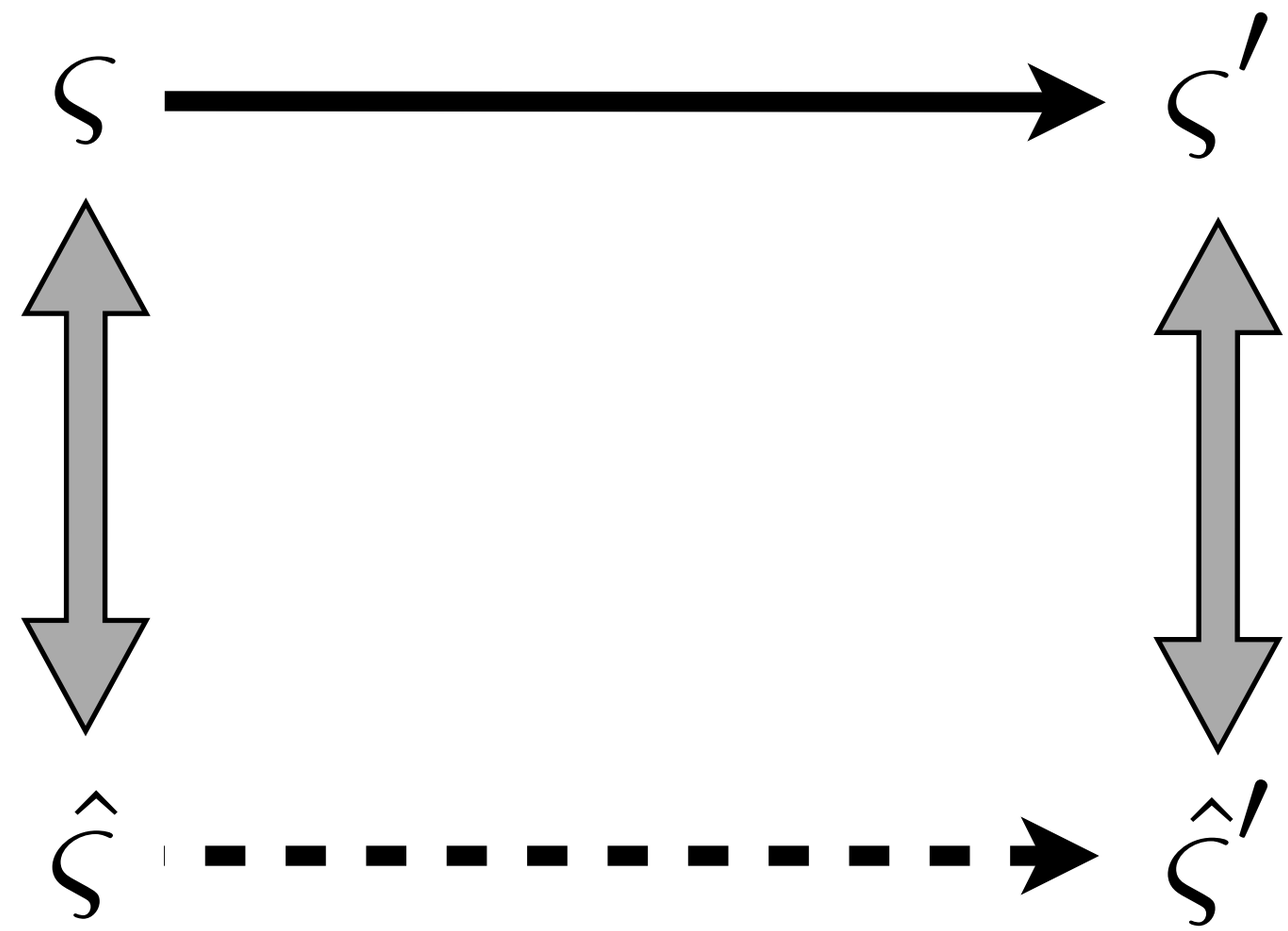
$$\langle \texttt{invoke-direct}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \mathcal{M}(id), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{a}_\kappa), \hat{t}'\rangle$$
$$\text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, \hat{fp}') \mapsto \sigma(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})]$$
$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}]$$
$$\hat{fp}' = \widehat{alloc}(\hat{\varsigma})$$
$$\hat{a}_\kappa = \widehat{allock}(\hat{\varsigma})$$
$$\hat{t}' = \widehat{tick}(\hat{t})$$

$$\langle \texttt{invoke-virtual}(r_0, \ldots, r_n, id) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}, \hat{t}\rangle \longmapsto \langle \mathcal{V}(id, v), \hat{fp}', \hat{\sigma}'', \mathbf{fnk}(\vec{stmt}, \hat{fp}, \hat{\kappa}), \hat{t}'\rangle \text{ if } v \in \hat{\sigma}(r_0, \hat{fp})$$
$$\text{where } \hat{\sigma}'' = \hat{\sigma}' \sqcup [(0, fp') \mapsto \hat{\sigma}(r_0, \hat{fp}), \ldots, (n, \hat{fp}') \mapsto \sigma(r_n, \hat{fp})]$$
$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_\kappa \mapsto \hat{\kappa}]$$
$$\hat{fp}' = \widehat{alloc}(\hat{\varsigma})$$
$$\hat{a}_\kappa = \widehat{allock}(\hat{\varsigma})$$
$$\hat{t}' = \widehat{tick}(\hat{t})$$

$$\langle \mathit{unop}(r_d, r_s) :: \vec{stmt}, \hat{fp}, \hat{\sigma}, \hat{\kappa}\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle$$
$$\text{where } v \in \hat{\delta}(\mathit{unop}, \sigma(r_s, fp))$$

$$\langle \mathit{binop}(r_d, r_1, r_2) :: \vec{stmt}, fp, \sigma, \kappa\rangle \longmapsto \langle \vec{stmt}, \hat{fp}, \hat{\sigma} \sqcup [(r_d, \hat{fp}) \mapsto v], \hat{\kappa}\rangle$$
$$\text{where } v \in \hat{\delta}(\mathit{binop}, \hat{\sigma}(r_1, \hat{fp}), \hat{\sigma}(r_2, \hat{fp}))$$

Theorem



Rapid

A fatal exception 0E has occurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.

* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
  lose any unsaved information in all applications.

Press any key to continue _

Passwords.
Patches.
Popularity.
Protection.
Preparation.
Prayer.

Thanks!

matt.might.net

matt@might.net

@mattmight