

Under the Hood:

A Mainframe Vulnerability Management Playbook

Amy DeMartine, Principal Analyst, Forrester
Ray Overby, President, Key Resources, Inc.

Objective:

To Educate Organizations on Mainframe Vulnerability Management and why the Mainframe should be considered a key part of their Digital Business Ecosystem.

Organization's are trending toward hybrid environments that are driven by mobility, IoT, cloud services, multiple operating systems and third-party mobile applications. However, this is creating vulnerability blind spots, which will lead to heightened security risks. Your mainframe is part of this digital business ecosystem – by 2030 there will be 40 trillion mobile transactions per day and the mainframe will process 75% of those transactions. And yet, the mainframe is rarely discussed when an organization assembles their penetration testing and risk management teams.

While mainframes are arguably the most secure computer system, they still are not impenetrable. All systems come with weaknesses, and the mainframe is certainly no exception. We need to think of the mainframe the way we think of any other computing platform when it comes to security threats and vulnerability management. Individuals responsible for enterprise security need to discard their costly perimeter-based security strategies and focus on critical data with a Zero Trust approach.

In this webcast, **Amy DeMartine**, featured speaker and **Ray Overby**, President and Co-Founder, Key Resources, Inc., will be discussing mainframe security strategies and how to incorporate these strategies into your current security practices.

2018 Key Resources, Inc. Reproduction Prohibited



To Educate Organizations on Mainframe Vulnerability Management and why the Mainframe should be considered a key part of their Digital Business Ecosystem.

Key Takeaways:



01

Learn why vulnerability management is now a board-level issue.



02

Understand the risks of not including the mainframe in your organization's risk management system.



03

Learn how mainframe integrity breaches can undermine your security systems.



04

The differences between Penetration Testing and Vulnerability Management.

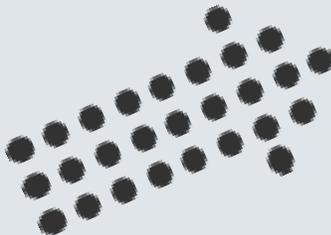


05

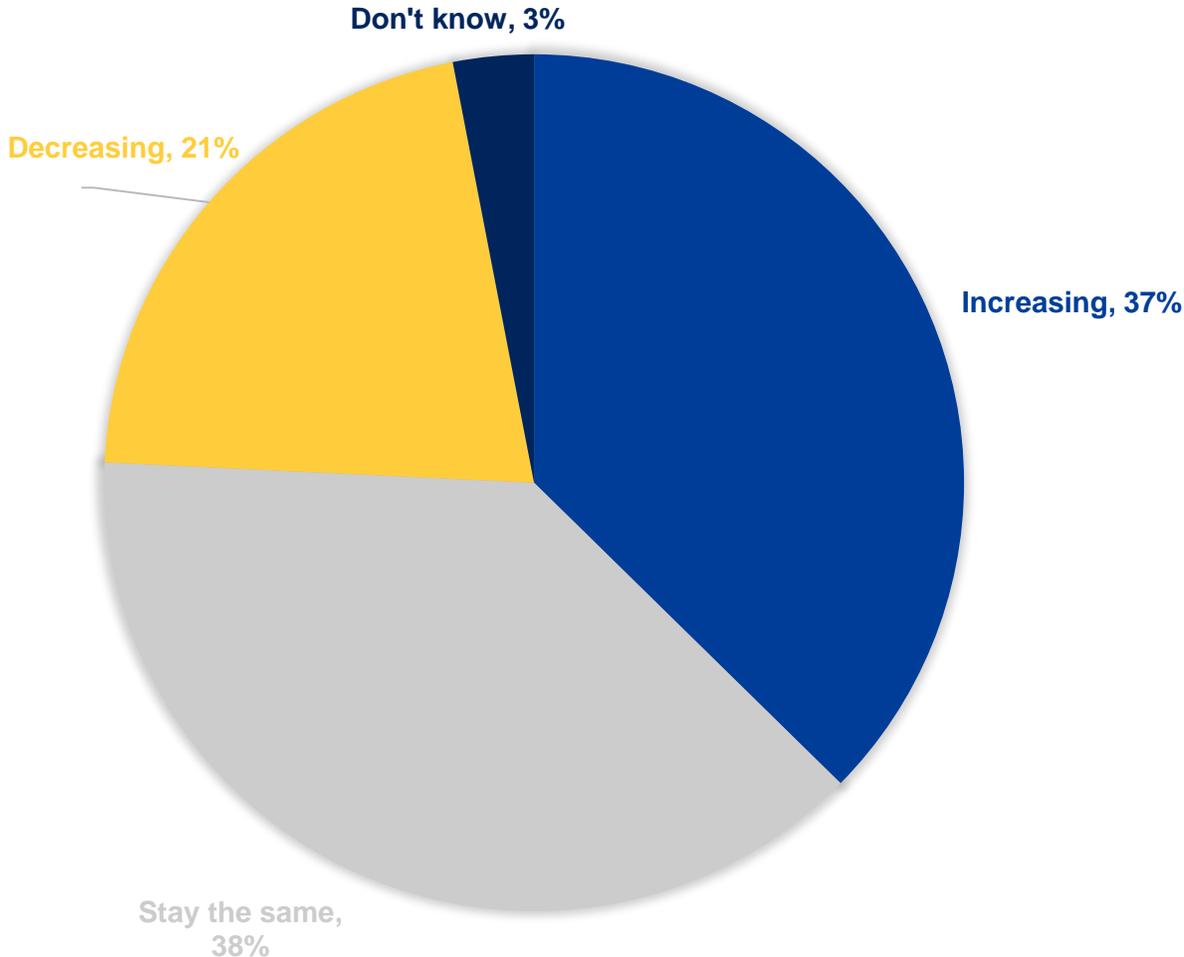
Why you should approach mainframe vulnerability scanning as a compulsory requirement versus a compliance requirement.

53%

Mainframe usage continue to increase.



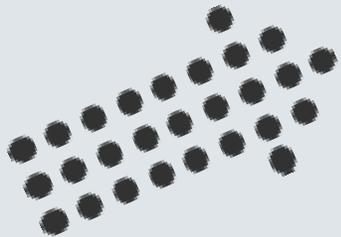
“In the next two years will you be increasing or decreasing your use of a mainframe?”



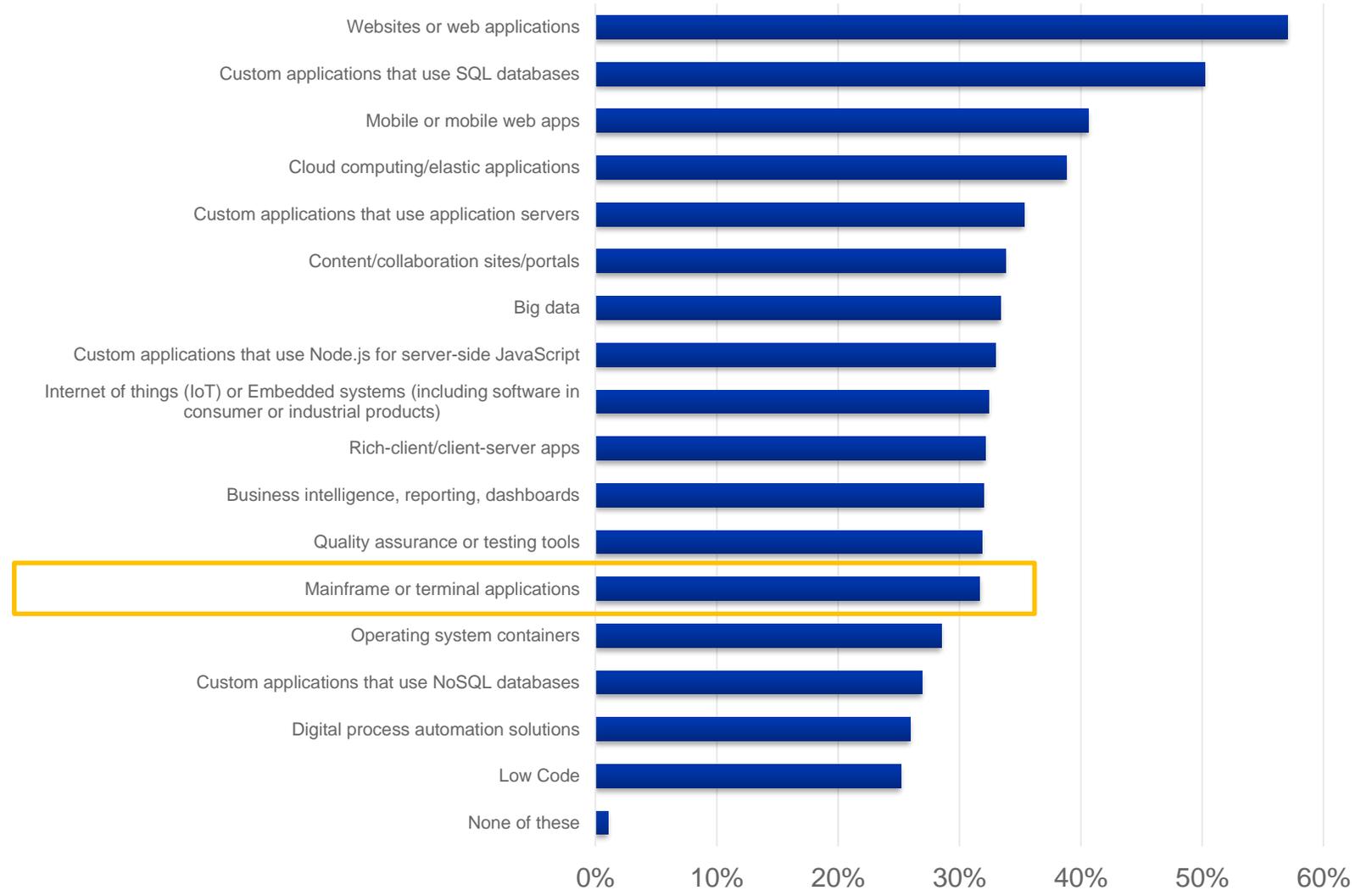
Base: 1340 Infrastructure technology decision-makers, Source: Forrester Data Global Business Technographics Infrastructure Survey, 2018

15.8%

Mainframe application development is alive and well at 32%.



“Which of the following types of software have you worked with in the past 12 months?”

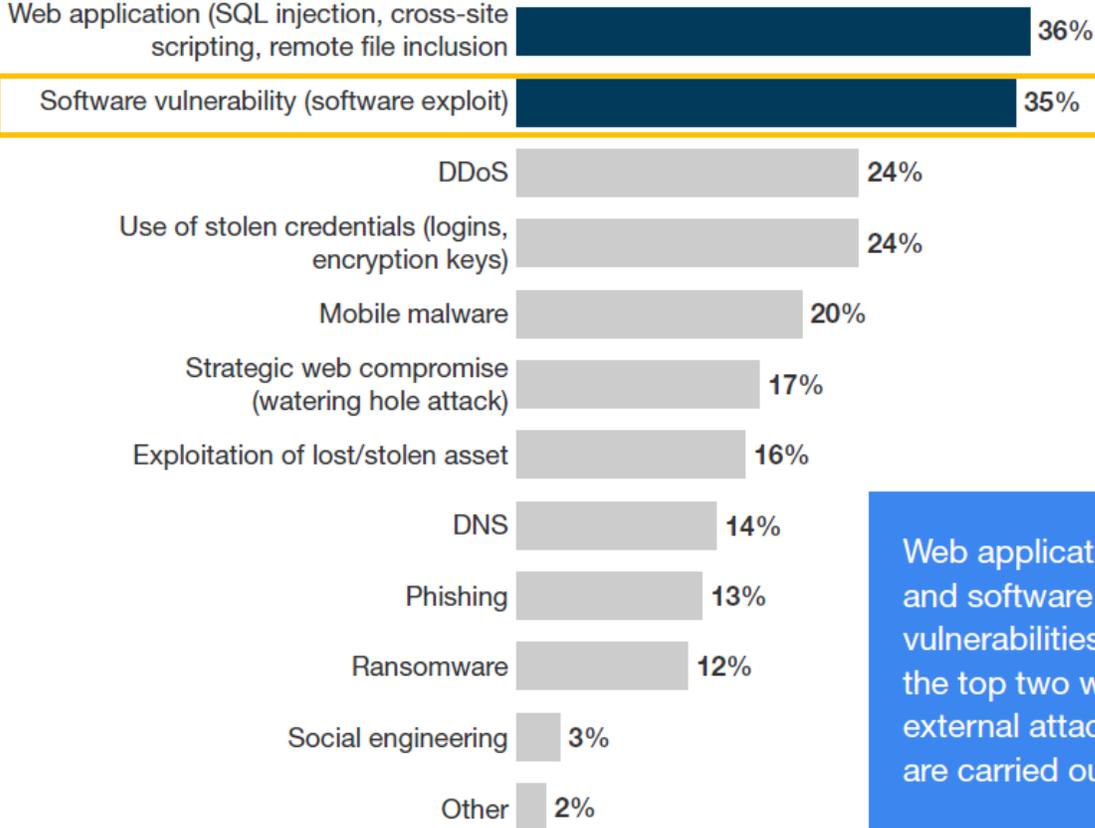


Base: 3228 developers, Source: Forrester Data Global Business Technographics Developer Survey, 2018

"Our mainframe applications **define what products we can offer. They drive **all** of our core business processes. They define how we invoice customers and recognize revenue."
(Senior executive, telecommunications company)**

40% of firms suffered a breach as a result of an external attack. This is how.

“How was the external attack carried out?”
(Multiple responses accepted)

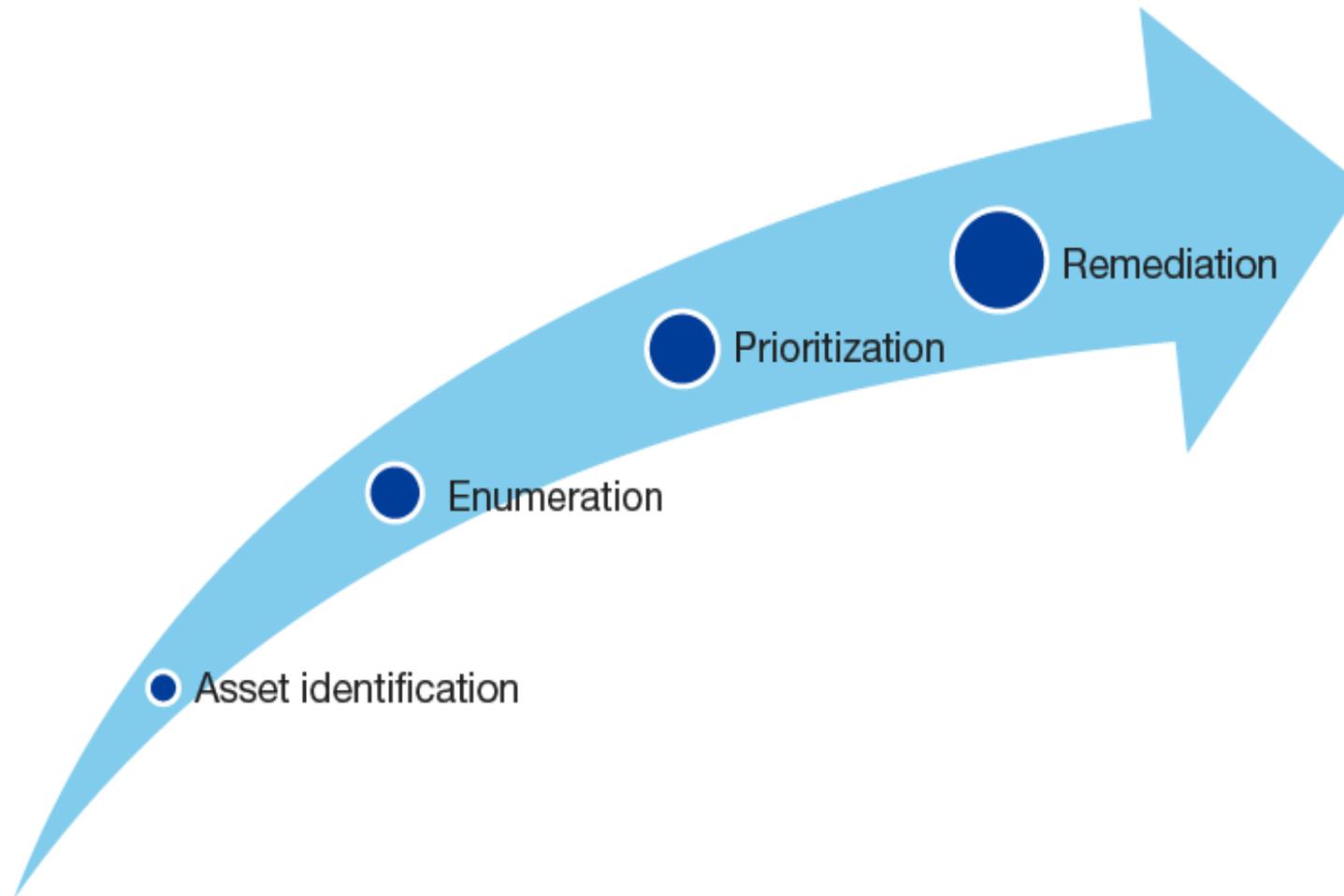


Web applications and software vulnerabilities are the top two ways external attacks are carried out

Base: 257 Network Path Security decision-makers who experienced an external attack when their company was breached

Sources: Forrester Data Global Business Technographics Security Survey, 2018

The Vulnerability Risk Management Process



What do you talk to your board about?



OUR NATIONAL DEBT:

\$14,255,583,167,035.

YOUR *Family share*

\$120,726.

THE NATIONAL DEBT CLOCK

Generate Metrics For Each Step

› Asset Identification

- Am I able to find all assets?

› Vulnerability Enumeration

- What is my coverage?

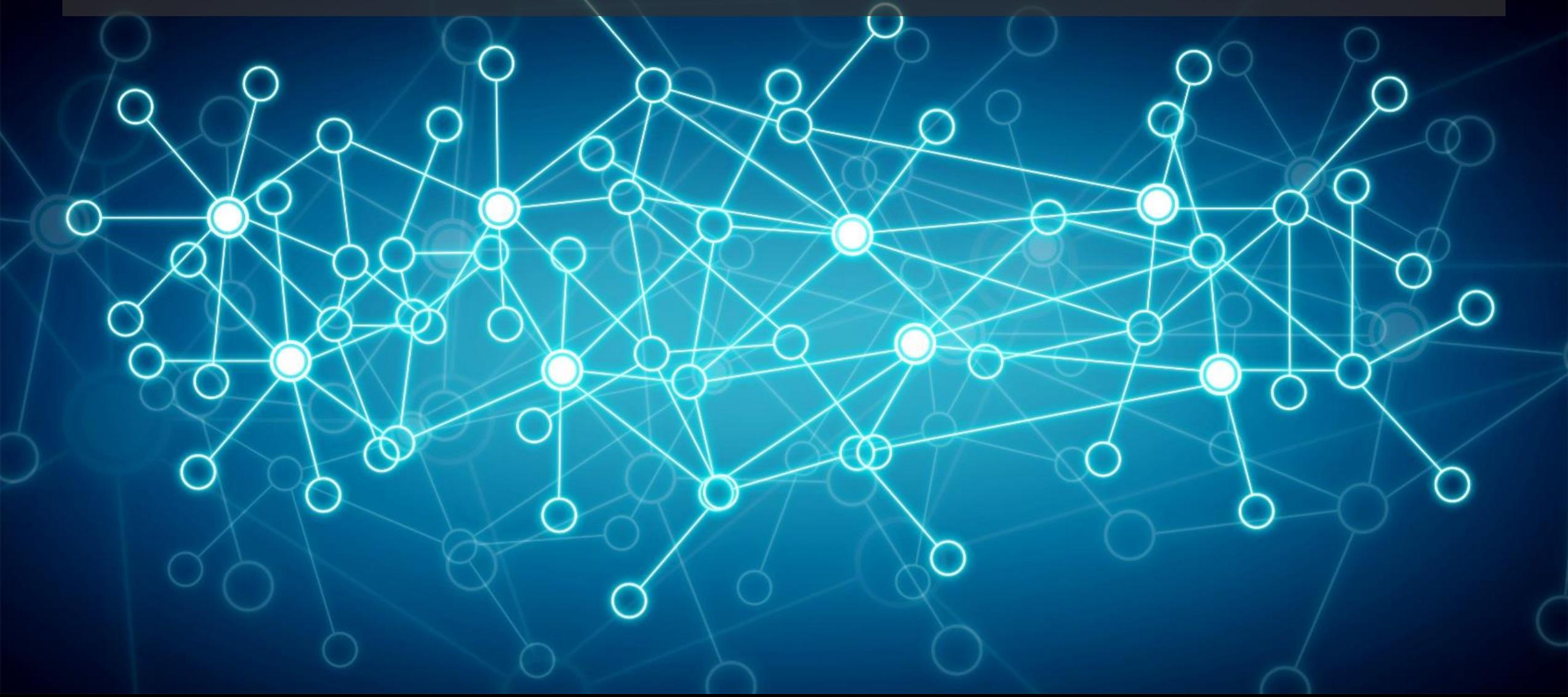
› Prioritization

- Do you have agreement on what is prioritized? Are exceptions the norm?

› Remediation

- Are my service levels based on asset and severity priorities combined? How good is my service level adherence?

What Is The Role Of Scanning In VM Process?



Are You Looking At Your Entire Assets?



Vulnerability Scanning Is Not Penetration Testing

Vulnerability Scanning

- Tools
- Can scan any asset
- Looking for known attack methods
- Run continuously
- Required by FFIEC; GLBA; PCI DSS
- In-house
- Comprehensive reports
- \$

Penetration Testing

- Manual with assist from tools
- Only covers assets that are exposed
- Exploiting weaknesses in the architecture
- Run periodically
- Required by FFIEC; GLBA; PCI DSS
- Outside services
- Details what was compromised
- \$\$\$

FORRESTER®

Amy DeMartine
ademartine@forrester.com

 @AmyDeMartine

Thank you

FORRESTER.COM

Introduction:

Are You Securing all of Your Assets?

-  So, how well is your mainframe secured?
-  Locked down your ESM (RACF, ACF2 & TSS)?
-  Rock solid process and procedures, JML, RBAC, Data Classification, Integration into your SIEM, etc, all sorted and done?
-  Happy, feeling secure?
-  Should you be?
-  Are you sure you have everything covered?
-  What if you have integrity issues, what could happen to you?
-  How about bypassing all of the controls you have in place?



The IBM z/OS Integrity Statement:

First issued in 1973, IBM's MVS System Integrity Statement, and subsequent statements for OS/390 and z/OS, has stood for over four decades as a symbol of IBM's confidence in and commitment to the z/OS operating system.

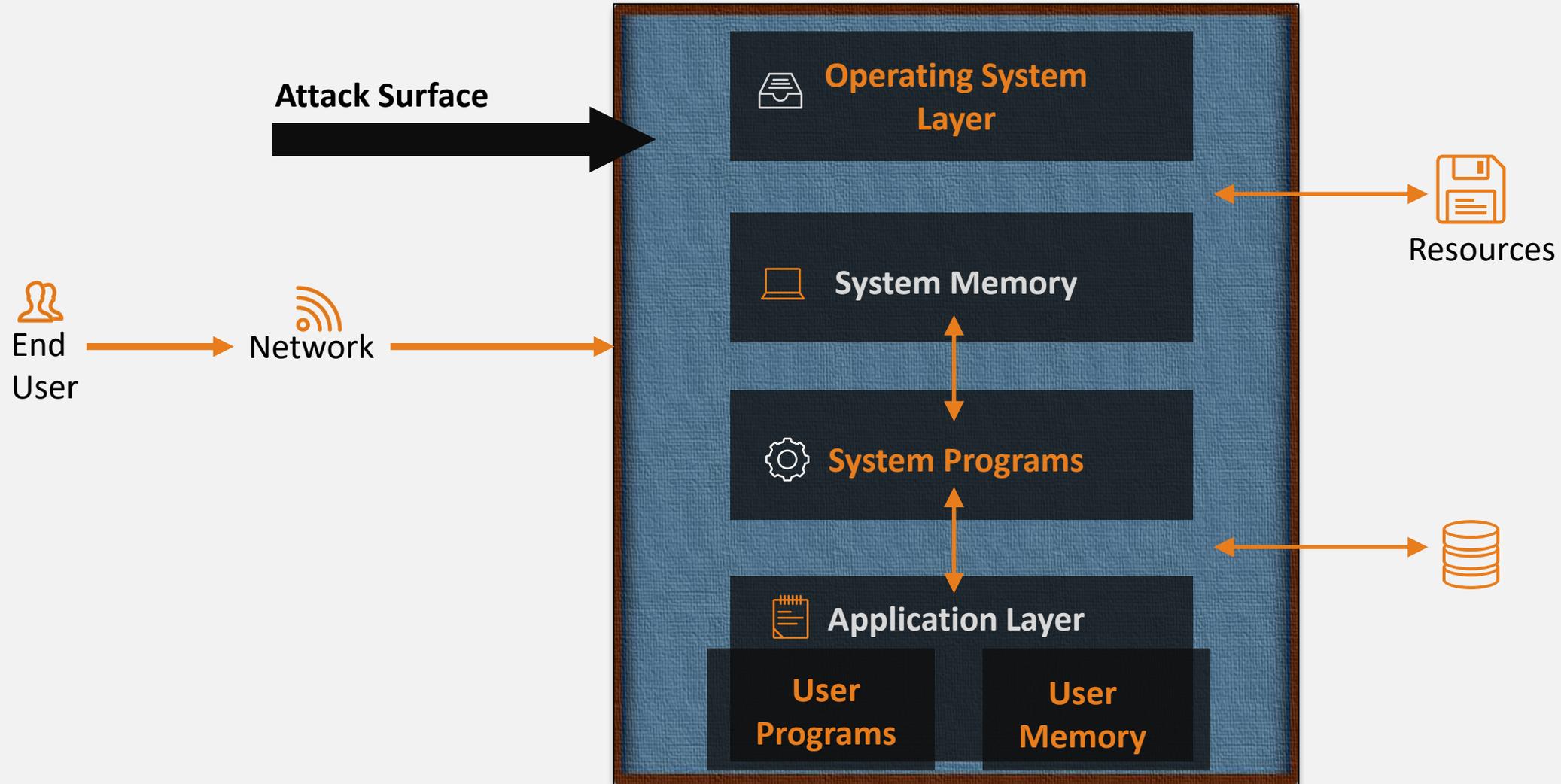
System Integrity is IBM's commitment, designs, and development practices intended to prevent unauthorized application programs, subsystems and users from bypassing system security—that is, to prevent them from gaining access, circumventing, disabling, altering or obtaining control of key system processes and resources unless allowed by the installation.

It allows authorization of system-level programs that need to modify or extend the basic functions of the operating system.



The z/OS Architecture:

What does Integrity Really Mean



Why is z/OS Vulnerable?



The attack surface is the boundary where attacks should be prevented.



With respect to z/OS Integrity, the attack surface is between user or non-authorized user programs and authorized system services:

Program Calls (PCs)
Supervisor Calls (SVCs)
Authorized Programs (APF)



These three interfaces are the methods used for requesting authorized system programs to provide services to a user program.



- APF Auth can be used to obtain Sup State and / or PSW Key 0 - 7.
- PSW Key 1 – 7 can be used to get to PSW Key 0. PSW Key 0 can modify any area of memory.
- Supervisor state allows the use of Privilege Instructions.

So how does a program break through the attack surface and a) get sup state or b) get PSW Key 0 – 7, or c) get APF auth?

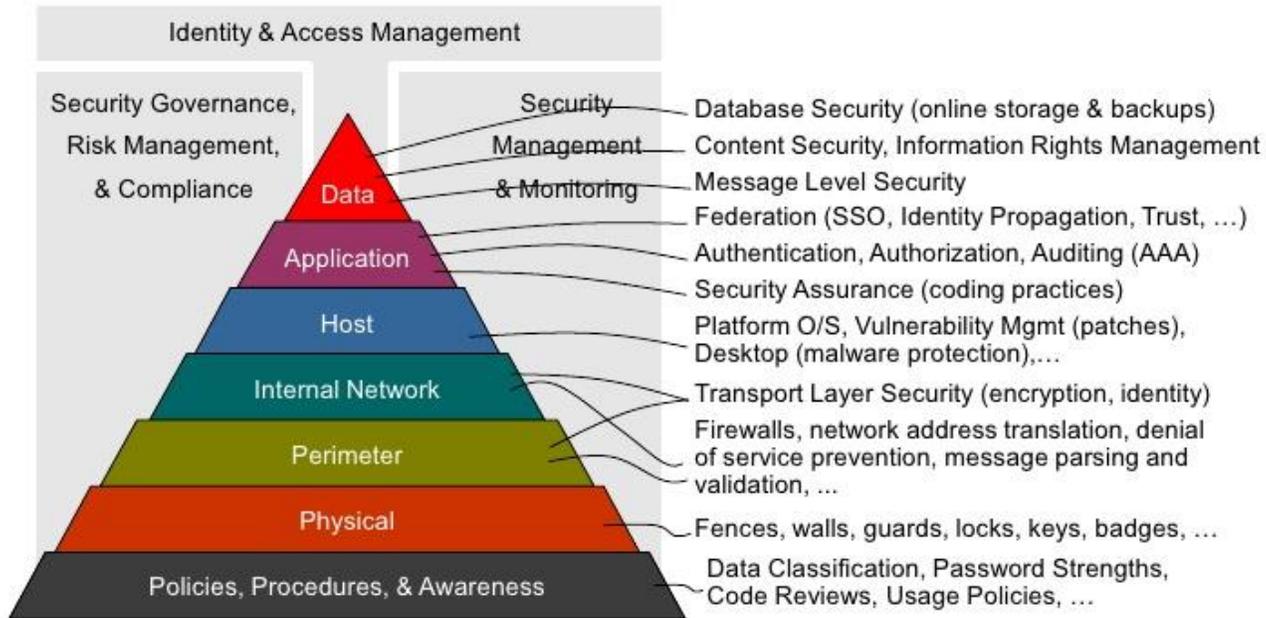
Typically, this occurs when one of the PCs, SVCs, or APF programs is either designed incorrectly or contains coding errors that allow a user program to bypass the integrity controls with a, b, or c above.

If a user program can bypass the controls in any of these methods with a, b, or c, it has broken through the attack surface and circumvents the IBM Statement of Integrity

How bad can it get: A rogue user program can deny availability by overwriting critical system areas causing the system to crash.

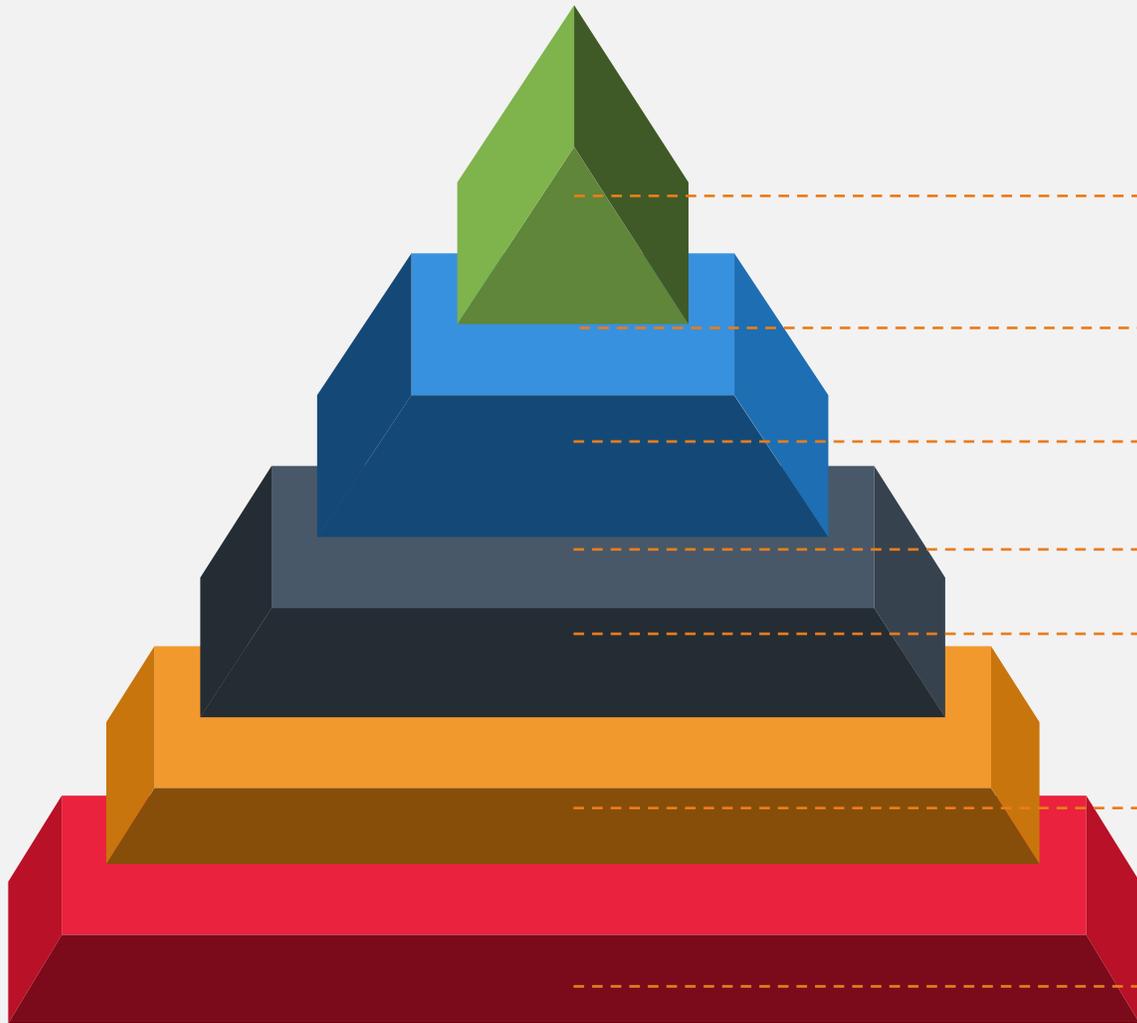
Vulnerability Management: A Look at the Distributed Model

Defense in Depth: Layers



Protect Information rather than Systems – use an Interactive Application Security Testing (IAST) approach to accurately find vulnerabilities in the OS layer that, when exploited, allow unauthorized and undocumented access to sensitive information.

z/OS Vulnerability Mgmt. A Conspiracy of Silence



Data

Application

Operating System

Internal Network

Perimeter

Physical

Policies, Procedures, and Awareness

**Why the
Gap?**



Mainframe Vulnerability **Compulsory** Management



Here is one from the wild

- ❖ Exploit implements dynamic privilege escalation.
 - ❖ Assigns RACF PRIVILEGED attribute to HACKER.
- ❖ The HACKER requires no extra-ordinary RACF privileges to execute the exploit.
- ❖ The HACKER is logged on to TSO on a z/OS 2.3 system with RACF as ESM.
- ❖ ACF2 or TSS would be compromised just like RACF
- ❖ The vulnerability exploit program is NOT APF authorized. It could be a CLIST or a REXX exec



Access the Dataset – ISPF 3.4

ISPF 3.4 Dataset List

Getting into Edit

```
Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching NOACCESS.TESTDSN Row 1 of 1
Command ==> CSR Scroll ==> CSR
Command - Enter "/" to select action Message Volume
-----
NOACCESS.TESTDSN UCBADF
***** End of Data Set list *****
```

Access Denied!

```
Workstation Help
EDIT Entry Panel
Object Name:
'NOACCESS.TESTDSN'
* No workstation connection
* Initial Macro . . . (Blank defaults to Type)
  Profile Name . . . (Leave blank for default)
  Format Name . . . (Leave blank for default)
  Panel Name . . . (Leave blank for default)
Options
- Confirm Cancel/Move/Replace
- EDIT Mixed Mode
- EDIT host file on Workstation
- Preserve VB record length
7 Warn on First Data Change
- ASCII data
Press ENTER to continue. Press CANCEL to cancel action.
```

```
ICH408I USER(NORMAL ) GROUP(SYSGROUP) NAME( )
NOACCESS.TESTDSN CL(DATASET ) VOL(UCBADF)
INSUFFICIENT ACCESS AUTHORITY
FROM NOACCESS.* (6)
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
IEC150I 913-38, IFG0194E, NORMAL, VATPROCO, ISP08597, 0ADF, UCBADF, NOACCESS.TESTDSN
***
```



```
Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching NOACCESS.TESTDSN
Command ==>
Command - Enter "/" to select action
NOACCESS.TESTDSN
***** End of Data Set list *****
E
```

User Could Not Access Dataset

```
EXPL0099 - EXPLOIT WORKED
***
```

Exploit Completed Successfully

```
Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching NOACCESS.TESTDSN
Command ==>
Command - Enter "/" to select action
NOACCESS.TESTDSN
***** End of Data Set list *****
e
```

Lets Try This Again

```
Menu List Mode Functions Utilities Help
ISPF Command Shell
Enter TSO or Workstation commands below:
==> call 'exploit1.load(expl0099)'
```

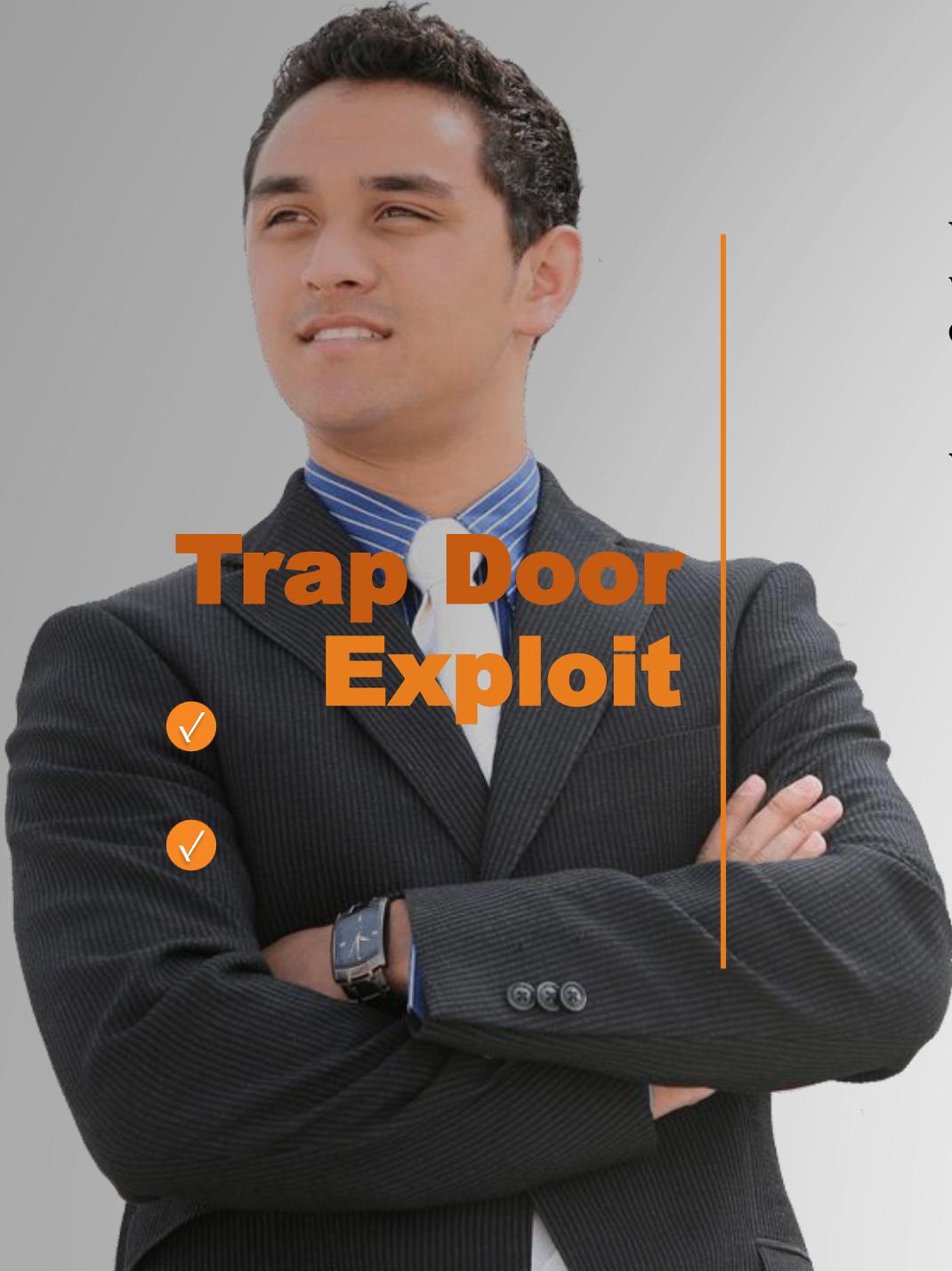
ISPF 6 – Run the Exploit

```
Workstation Help
EDIT Entry Panel
Object Name:
'NOACCESS.TESTDSN'
* No workstation connection
Initial Macro
Profile Name
Format Name (Blank defaults to Type)
Panel Name (Leave blank for default)
Options
- Confirm Cancel/Move/Replace
- EDIT Mixed Mode
- EDIT host file on Workstation
- Preserve VB record length
7 Warn on First Data Change
- ASCII data
Press ENTER to continue. Press CANCEL to cancel action.
```

Getting Into Edit

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT NOACCESS.TESTDSN Columns 00001 00072
Command ==>
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 No one should have access to this dataset.
***** Bottom of Data *****
```

User Now Has Access



Trap Door Exploit



You just saw an example of an exploit of a Trap Door vulnerability. The exploit does not require APF. The exploit could be a CLIST or a REXX exec.

What did We Demonstrate:

- ✓ *Demonstrated that the user does not have access to the dataset before the exploit program was executed.*
- ✓ *Demonstrated that the user now has access to the dataset, and no security logging occurred after the exploit program was executed.*

Trap Door Exploit – What to Remember



- ❖ A Trap Door vulnerability is ALWAYS exploitable!
- ❖ The user is logged on to TSO on a z/OS 2.3 system with a RACF Userid that has no extraordinary security authorities.
- ❖ The exploit dynamically elevated the RACF credentials of the user.
- ❖ With slight changes the exploit would work with ACF2 or TSS
- ❖ It gives the exploiter the RACF PRIVILEGED attribute
- ❖ RACF PRIVILEGED bypass's RACF authorization checking (including logging).
- ❖ This is a typical Trap Door exploit of an OS Layer Vulnerability found in z/OS operating system code that has been found on production mainframes.
- ❖ z/Assure VAP CVSS score for a Trap Door vulnerability is 8.2-8.6.

Based on Facts Our Premise: The z/OS Operating **System Layer is Vulnerable**

It's All About
Integrity

System integrity is the reason mainframes are the most secure computer platforms, but vulnerabilities in code can breach integrity without warning

You can't trust vendors (distributed or mainframe) to provide secure software, and you can't rely on the software community to actively or organically surface vulnerabilities

What Does
That Mean?

z/OS is Vulnerable

Vendor software builds and releases are hurried with less testing; developers do not always have the skillset necessary to write integrity based software. Let's face it. Software has holes. And hackers love to exploit them. Mainframes are the big fish!

01

Without Integrity you cannot have security; vulnerabilities in the OS layer can breach integrity without warning.

02

ESM's: RACF, CA ACF2, and CA Top Secret are essential for establishing permissions and access control, but they were not architected to protect against integrity vulnerabilities

03

IBM's System Integrity architecture is the reason mainframes are highly secure computer platforms, but vulnerabilities in OS level code will allow breaches (without the Enterprise Security Manager (ESM) issuing any type of warning).

04

Mainframe Data Security should be built around a strong inclusive Vulnerability Management Program; NOT just around ESM's.

05



What about Malware and Ransomware?

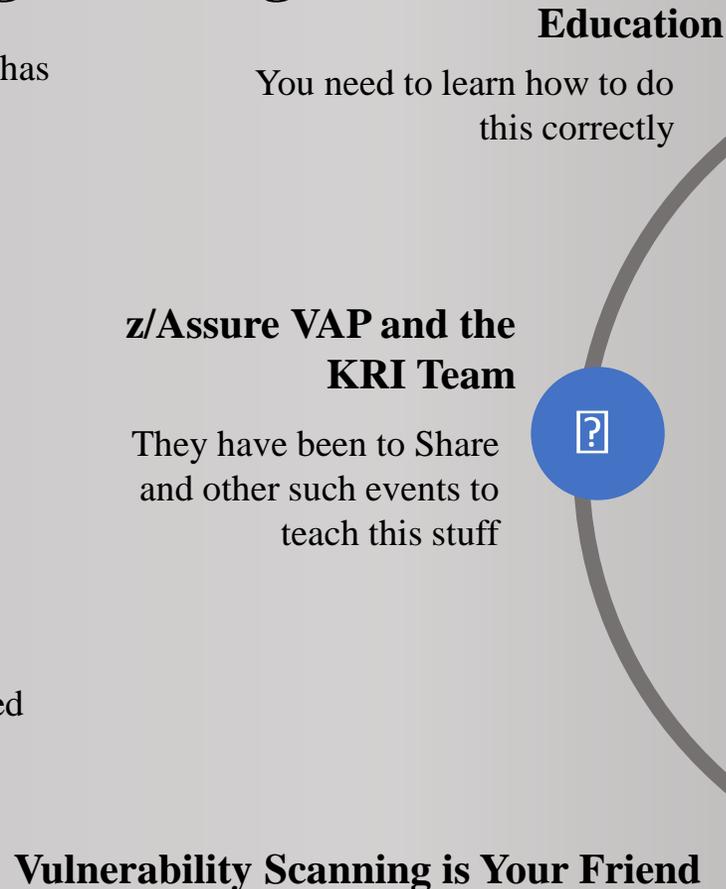
- ❖ Asked this question by some C-Levels: “Do you think our mainframe could ever be infected by ransomware?”
Answer: YES!!!!
- ❖ Just another Program; Polymorphic
- ❖ Ransomware is comprised of three major parts:
 - ❖ Infection vector (phishing, web drive by, social engineering)
 - ❖ Payload - generate key, enumerates and encrypts files
 - ❖ Command and Control (optional)
 - ❖ Phones home
 - ❖ Communicates with victims
 - ❖ Stores keys
 - ❖ Other items as required (e.g. customer service)
- ❖ Core to Ransomware is Crypto; good strong and fast Crypto!

How to do it right!

What could happen if it goes wrong?

- Lets suppose we have a piece of code that has either:
 - ❖ Been supplied by a Vendor/ISV
 - ❖ Been written by your in-house technical teams
- However, the code has a vulnerability that can be exploited
- What could happen?
- Bypass all of your ESM controls irrespective if you run RACF, ACF2 or TSS
- Bypass/Disable all the logging and monitoring controls you may have deployed
- Encrypt all of you data (Production, Development & System)

How to do it right!



What is z/Assure® VAP

2011 - 2018 Key Resources, Inc. All Rights Reserved.

OSIT™ – Operating System Security Testing

- ❑ IAST architecture – we call it OSIT. Scans leaves the code untouched, and does not require programmer involvement. This is not a checklist of known vulnerabilities!
 - ❑ z/Assure VAP is a Binary code scanner.
 - ❑ Batch Jobs are submitted to observe the code as it is executing,
 - ❑ Accurately establishes whether a vulnerability exists.

z/Assure VAP Reports using the CVSS Standard

- ❑ Classifies the source and types of z/OS code vulnerabilities found and provides a CVSS score for each vulnerability.
- ❑ Developers know exactly where to go to fix the vulnerability (generates a detail report of the offset into program where the vulnerability is located).
- ❑ Management knows the immediate risk associated with each vulnerability (Classification = Clarity).

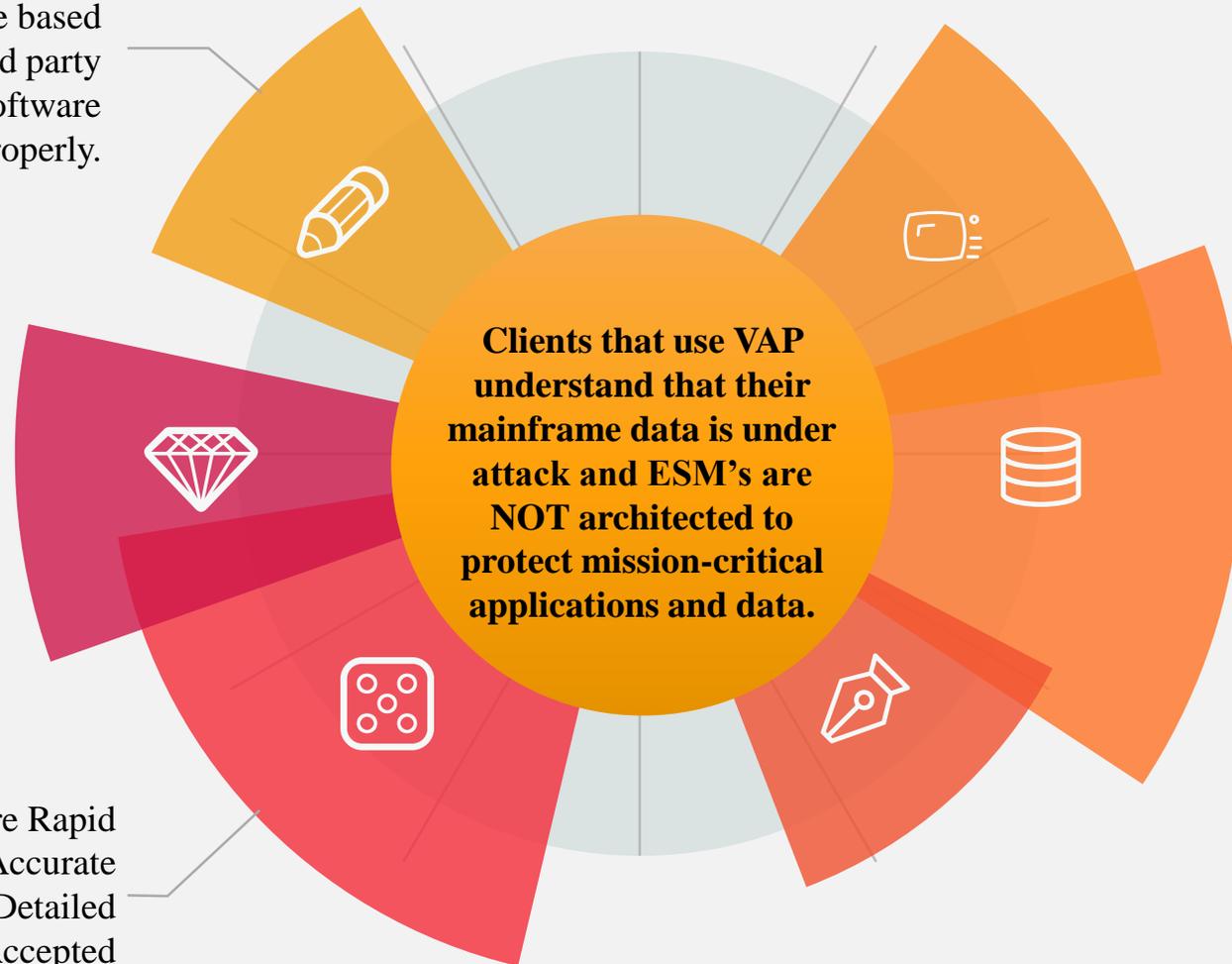
A Sample of the Categories of Vulnerabilities z/Assure VAP Identifies

- ❑ Storage Alteration
- ❑ Trap Door
- ❑ System Instability
- ❑ Least Privilege
- ❑ Storage Reference
- ❑ Identify Spoofing

- ❑ Categorization = Clarity

Reasons our Clients use z/Assure Vulnerability Analysis Program (VAP)

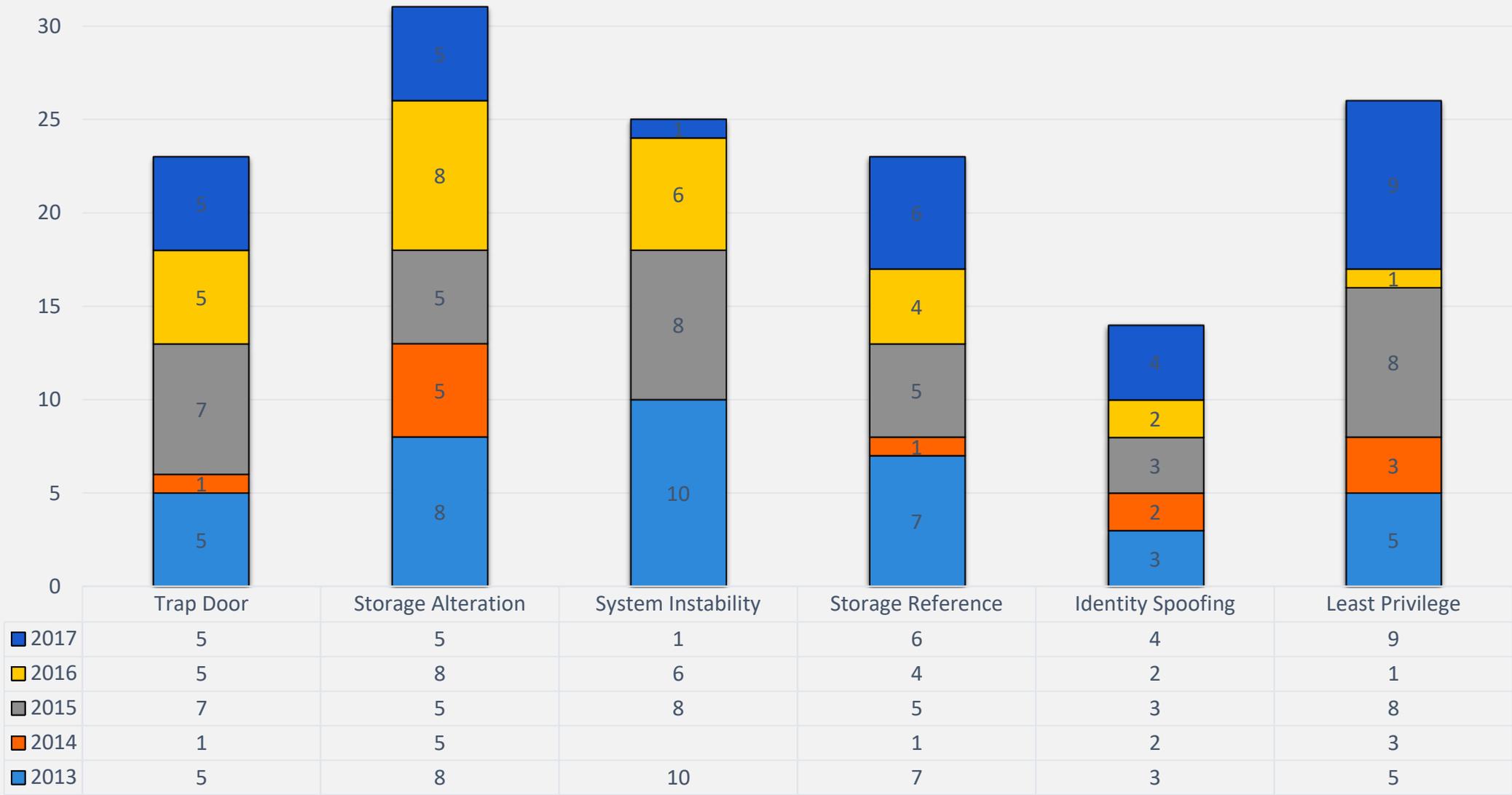
The number of code based vulnerabilities is growing as third party vendors continue to develop software that is not tested properly.



Data can be compromised and the compromise goes undetected.

Results are Rapid
Results are Accurate
Results are Detailed
Results are Accepted

z/OS Integrity Based Zero Day Vulnerabilities Found by z/Assure VAP



2013 2014 2015 2016 2017



A Mainframe-based Vulnerability Management Program should include **ALL** layers.

Mainframe Data Security should be built around a strong Vulnerability Management Program; **NOT** just access and privilege management.

ESM's are essential for establishing permissions and access control, but this is **NOT** a complete security solution.

Secure your environment at every level. Make mainframe OS-level integrity a part of your overall security strategy.

Vulnerability Management across **all** platforms and operating systems is now the standard for many international compliance programs.

It should be noted that the IBM z/OS Statement of Integrity only applies to IBM code. It does not apply to any ISV code or installation written code. You, the z/OS system owner, are responsible for verifying the integrity of any code you add to z/OS.

<https://www.ibm.com/it-infrastructure/z/capabilities/system-integrity>.

Update your Vendor contracts to make sure they are responsible for fixing Integrity Vulnerabilities in a reasonable amount of time. Vendors should not be able to fall back on "product will need to be re-architected".

Integrity based code vulnerabilities can suppress forensic evidence leaving nothing for security professionals to review.

KEY

RESOURCES
The Key to zSystems Integrity

thank you!

Ray Overby
ray.overby@krisecurity.com