**EMCS2000: Introduction to Computer Security :** Post-Work: Module 2 Report **( User Guide )**

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

### Section 1

#### Scenario

Browser security at NASA is extremely important, but at the same time extremely hard to get right. As an organization, we want to be smart about developing Assurance for users by creating Policies and provisioning Privileges that Protect everyone while preserving performance.

#### Security Objectives

Our objective is to develop a NASA Browser that uses Artificial Intelligence to create dynamic use policies based on real-time threat intelligence. As a user this dynamic policy creation, provisioning and enforcement should be seamless. As a benefit of using the new NASA Browser, code-named **Jemison**, you will notice fewer roadblocks in form of analog permission requests, fewer restrictions on the range of sites you can visit and overall less worry about visiting suspicious websites.

#### Scope

This new Browser will be deployed through the entire organization for all user on Desktops, Laptops and even mobile devices. The browsers on TV’s and IoT devices will be disabled, and no other Browser other than this specially developed Browser will be allowed.

#### Technical Description

**Jemison** will be much like the popular Chrome Browser you may be familiar with because it is built with the same code base as Chrome. Google Chrome is derived from an open source project called Chromium. The source code for Chrome is proprietary to Google, but the Chromium Open Source Project that shares over 98% of the same code is open source. We have engineered changes in the code that controls policy engine in Jemison, adding an Active Policy Agent, powered by Artificial Intelligence and updated by a private BlockChain. This new policy engine will make real-time decisions about the safety of websites, actions you take in the browser and files you open with the browser. And since Jemison is powered by AI she can also help pick strong passwords that are easy to remember. Here’s an example of Jemison helping a user with password creation:

**Jemison**: “Looks like you are choosing a password, would you like some help?”  
**User**: Sure.  
**Jemison**: “Great, think of a line from your favorite movie or song. Don’t write it down, just think of it. Choose something you know like the back of your hand”  
**User**: Ok, got it.  
**Jemison**: “Ok, now use the first letter of each word in that phrase to create your password, capitalizing words like names or key points … then add a small number and a special character that you can remember. For example: ‘Show me the Money!’ -Jerry Maguire … would be ‘*SmtM!JM747#!*’ ”

Jemison’s job is to create smart policies that preserve your privileges as a NASA employee and protect everyone in the organization from malicious websites.

#### Troubleshooting

If you are having problems with Jemison, sending feedback to the Help Desk will be easy. Simply open a feedback ticket inside the browser and Jemison’s AI will re-evaulate your permissions right on the spot. In some cases, Jemison will have to route your ticket to the Help Desk, but hopefully once for each type of issue. We are training Jemison to find problems before they occur so people who might encounter the same issue won’t need to file a ticket. Your ongoing feedback makes Jemison smarter while helping to keep NASA safe.

#### Exceptions

Using the new browser will be enforced across the organization and across teams. The only exception may be the QA team for purposes of benchmarking performance against a browser that does not have the enhanced security features.

Users are not required to use the new browser on personal devices unless they want to use their personal device on the corporate network.

Section 2

#### Rationale

Web Browsers are a popular surface for attacks because they may provide an open door to the World Wide Web, most can execute code in a powerful Virtual Machine and yet their regular use tends to lull users into a relaxed state, one in which they lower their guard. Even the most vigilant user can miss most of the I/O happening in the browser’s ecosystem, ie. cookies, cache, files sitting the downloads folder and more. Most of the software that protects endpoints does so by trying to intercept URLs but cannot protect against code that executes in a browser session once the user is already on a site. By redefining the way a browser's makes policy decisions and leveraging the power of Artificial Intelligence, we can proactively find threats before they happen and modernize the way we safeguard users.

**EMCS2000: Introduction to Computer Security:** Post-Work: Module 2 Report

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

#### Section 1

##### **Scenario**

Network Security and EndPoint Security both depend heavily on the concept of **Assurance**. Assurance means enacting the proper ***Policies, Permissions, and Protections*** to make browsing safe.[[1]](#footnote-0) In order to gain this assurance and enact effective policies, organizations should have insight into the inner workings of browsers and how they handle code running inside of their application walls. At NASA the browser is the least secured piece of software in the application stack. We need a smart scalable solution, one that protects users in real time, without overwhelming the security team with analog permission requests or bogus alerts.

##### **Security Objectives**

Our objective in this security plan is to create real-time rules for browser behavior that apply to a wide range of activity, are remotely enforceable and utilize machine learning to make real-time decisions about security. In this document, we will focus on:

* **Describe** the type of threats we need to secure against
* **Investigate** how the Chrome Browser works and provide an introduction to the Chromium Open Source Project
* **Outline** how we can make new, safer Browser upgrading the Chromium Process Manager
* **Propose** ways we can measure success and implement effective threat modeling

This document will primarily focus on the types of threats we need to protect against and how we can engineer a solution. It will not address legal or budget concerns.

##### **Scope**

This security plan will have a sharp focus on Chrome and the way it handles browser activity and security. Chrome, one of the most popular browsers is based on an Open Source Project, Chromium. While the Chrome source code is not open, the code that is used to build it is. This section provides a scope for our security development roadmap.

###### General Target

**Browsers** are an application launchpad, an application container and the door to data storage. Desktops, laptops, phone, tablets and even some TVs have browsers. This security plan and the development efforts will include all devices that contain a Browser, or the capabilities to have a browser.

###### General Audience

**Users** operate browsers and their actions in the browser may cause the organization to be vulnerable to attacks. We need a way to **allow most actions**, but **limit the scope of these actions** so mistakes don’t lead breaches. Therefore our security plan will focus on creating smart dynamic policies around user behavior in the browser, policies that protect users and do not impede their work.

###### Affected Systems

**Browser Navigation Rules** or decisions about websites users can visit are usually handled by **endpoint protection**. “*The components involved in aligning the endpoint security management systems include a Virtual private network (VPN) client, an operating system and an updated antivirus software.*”[[2]](#footnote-1) However, there are many instances when the user intentionally or unintentionally turns VPN and antivirus OFF, breaking most traditional endpoint detection. The solutions discussed here will seek to protect the user, even in these two layers of protection are gone.

###### Development Stakeholders

**Product Managers from Security Operations** will be responsible for creating the product requirements.

**System Engineering** will be responsible for the technical requirement and the developing solution.

**Quality Assurance** will be responsible for developing acceptance criteria and performing UAT.

##### **Technical Description**

Chromium development starts with understanding the Chromium code base. First of all, Chromium is not the Chrome Browser itself. The [Chromium Open Source Project](https://chromium.googlesource.com/chromium/src/), much like the Mozilla Project, is supported by the community. Also, “Chromium's multi-process architecture is a radical departure from other web browsers.” [[3]](#footnote-2) This multi-process architecture makes each tab its own process managed by the Chromium Central Process Manager. Our approach is to leverage this architecture to create dynamic policy enforcement while preserving the user experience and performance.

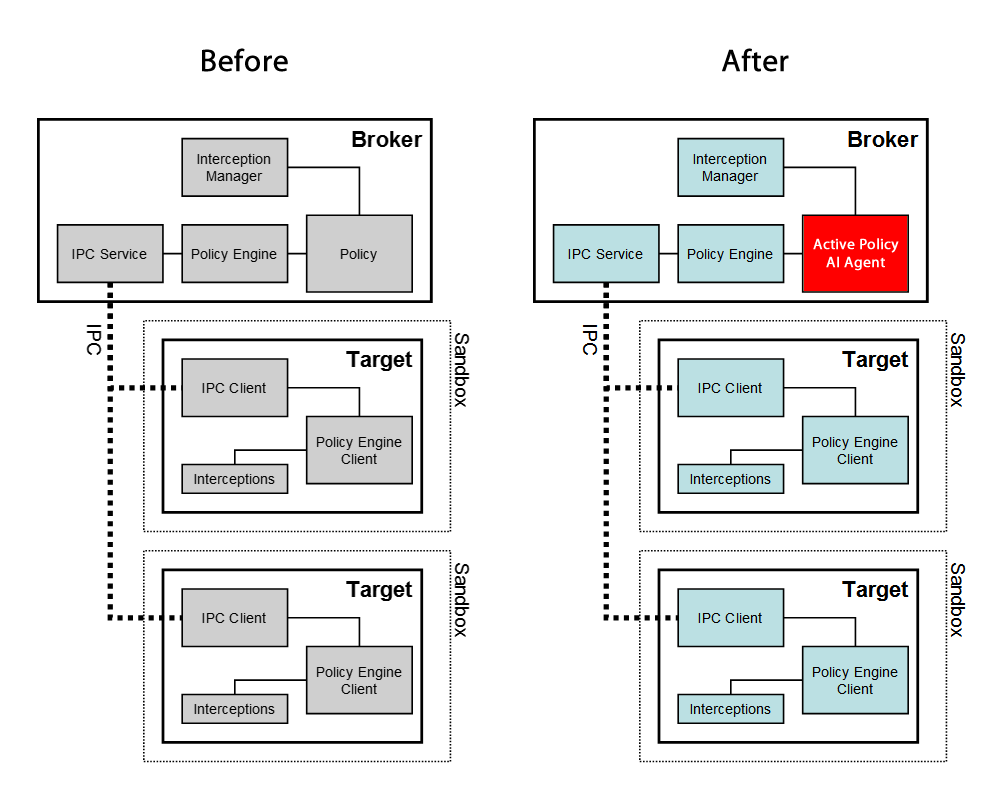
###### Chromium Central Process Manager

Separate processes for each tab helps achieve a number of amazing innovations. First, the code running in that tab’s process also has its own thread and its own memory block. Here is an overview of the sandboxing architecture :

*“Given the renderer is running in a separate process, we have the opportunity to restrict its access to system resources via sandboxing. For example, we can ensure that the renderer's only access to the network is via its parent browser process. Likewise, we can restrict its access to the filesystem using the host operating system's built-in permissions.  
  
In addition to restricting the renderer's access to the filesystem and network, we can also place limitations on its access to the user's display and related objects. We run each render process on a separate Windows "Desktop" which is not visible to the user. This prevents a compromised renderer from opening new windows or capturing keystrokes.”[[4]](#footnote-3)*

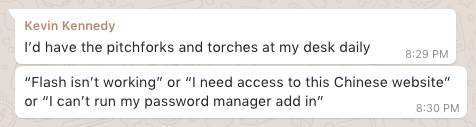
Chromium’s architecture mimics the type of isolation that processes are given when running in a modern Operating System. This type of architecture is ideal for creating and enforcing custom security policies. If we can actively control the policies used to make decisions in the browsers and dynamically grant permissions to processes running inside tabs, we can do a better job at protecting the users from harming themselves or the organization without impeding access to legitimate sites or slowing down performance.

Google built Chromium with security in mind, however, the policy management features are a little outdated, slow, and relatively weak. Google pointed out this weakness in an [online comic](https://www.google.com/googlebooks/chrome/big_32.html) they used to announce the Chromium project in 2008. The security API is a public, however, and they did this with the idea that the community would improve on the design. Our idea is to create an **Active Policy Agent AI Agent** in the Broker module, one ideally updated by private blockchain like the one in BETA at Oasis Labs. The policies in the browser could be updated from a reliable source in real time while the AI module makes decisions about safety based on derived variants of the threats before they are found by threat researchers.



[[5]](#footnote-4)

The Active Policy AI Agent operates under the basic principles of [Threat Hunting](https://en.wikipedia.org/wiki/Cyber_threat_hunting), is updated by a trusted source using Private BlockChain [Smart Contracts](https://en.wikipedia.org/wiki/Smart_contract) and relay Threat Intelligence using BlockChain back to the security team for analysis, alerts, and actions. When I described some early ideas of this system to a classmate their initial response was :



With the Active Policy AI Agent and a system that relays the behavior of users back to the Security Team there would be no need for “pitchforks and torches.” Making decisions about a particular Chinese website, or a certain version of the Flash Player based on real-time threat intelligence and the actual processes running in the tabbed sandbox of a user’s browser eliminates the need for most analog “permission granting.”

Using Oasis Labs Testnet and a fork of the Chromium Project, I’ve started development of an Active Policy Agent AI.

###### Passwords

The Active Policy AI Agent doesn’t just need to scan for malicious code or suspicious websites, it can be configured to ensure that passwords are strong and memorable. The best passwords are long, include several different types of characters and most importantly **can be remembered**. Forcing users to pick long passwords that they can’t remember leads to a much worse CyberSecurity no-no, saving them in a file called “passwords.” Quoting myself:

*“We all remember lyrics from songs, or lines from movies, or famous quotes. If we intersperse these with numbers and special characters we can make a pretty strong password. For instance, many people know this quote: "Hello, my name is Inigo Montoya, you killed my father, prepare to die." What if we converted that to HmniIMykmfPTD747#! ... Here I just used the first letter of each word in the phrase and capitalized the Hello, Inigo Montoya and Prepare To Die, then added 747, hash, bang.”[[6]](#footnote-5)*

The Active Policy AI Agent will sense when the user is picking or using a password and can provide prompts that help the user create a strong password using the method above, almost like an Alexa service.

**Policy Agent**: *“Looks like you are choosing a password, would you like some help?”*

**User**: Sure.

**Policy Agent**: *“Great, think of a line from your favorite movie or song. Don’t write it down, just think of it. Choose something you know like the back of your hand”*

**User**: Ok, got it.

**Policy Agent**: *“Ok, now use the first letter of each word in that phrase to create your password, capitalizing words like names or key points … then add a small number and a special character that you can remember. For example: ‘Show me the Money!’ -Jerry Maguire … would be* ‘**SmtM!JM747#!**’ ”

##### Troubleshooting

###### Approach and Methodology

Troubleshooting in this security plan means uncovering threats before they happen and avoiding false positives. Nobody wants to use a Browser that is constantly impeding their browsing experience, and yet the security team wants to stay two steps ahead of attackers. By adopting the Threat Hunt Model we can analyze browser activity and use the findings to enforce policies for users across the organization. However, threat hunting is not enough. To be successful we need to create standards around communication and make the threat hunt model a mindset.

The traditional definition of Threat Hunting is *“the process of proactively and iteratively searching through networks to detect and isolate advanced threats that evade existing security solutions.”*[[7]](#footnote-6) The problem in most modern security plans is that the threat hunters, system engineers, and help desk agents don’t communicate. If we imagine each of these positions as a part of the human body, white blood cells, red blood cells, and the sensory system respectively, we can imagine how the lack of communication can lead to massive dysfunction. By integrating AI into the browser as a policy agent, we are effectively **making the browser a threat hunt agent**, which is powerful. However, if we fail to communicate across teams we will still fail to take action when it is most needed.

Developing the capacity to troubleshooting the system and its effectiveness involves a radical paradigm shift. If all of the people described above take on the mentality of a threat hunter, the level of organizational vigilance is raised exponentially. There is broad agreement in the Threat Hunting community that effective classification leads to better data analysis. As a former Pandora engineer, I agree. Proper classification makes predictive systems exponentially stronger, much like a twined rope.

###### Implementation

Here are some concrete actions, policies, and procedures we can take to improve troubleshooting in the threat hunt paradigm.

**Help Desk Ticket Cross Analysis** will help us understand a user’s request in context. A simple request for privileges, or access to a site can be harmless when evaluated in isolation. Help Desk Agents should have access to the real-time analysis captured by the Active Policy Agent AI and use this data to make better decisions about access and permissions.

**Threat Modeling** is a well know practicing but the results of this research are not always distilled down to policies that can be consumed by machine learning engines. As a part of increasing the effectiveness of the Active Policy Agent AI, the security team will work hand in hand with Data Science to train new models as they surface in the threat landscape.

**Regular Joint Cyber Security Debriefing** with representatives from SecOps, Engineering, Human Resources and each Business Unit (Research, Finance, Operations, etc ) should be held on a bi-weekly basis. The purpose of the meeting should be to share information about threats and help reinforce CyberSecurity principles.

**Feedback Mechanisms** in the Browser should be at the ready and easy to use. When users have problems with the Browser they should be able to quickly report issues. This will also include crash reporting. The best feedback mechanism only requires a few clicks, captures the user sentiment and relays a snapshot of the Browser’s log back to the Engineering Team.

##### **Exceptions**

Using the new browser will be enforced across the organization and across teams. The only exception may be the QA team for purposes of benchmarking performance against a browser that does not have the enhanced security features.

Users are not required to use the new browser on personal devices ***unless*** they want to use their personal device on the corporate network.

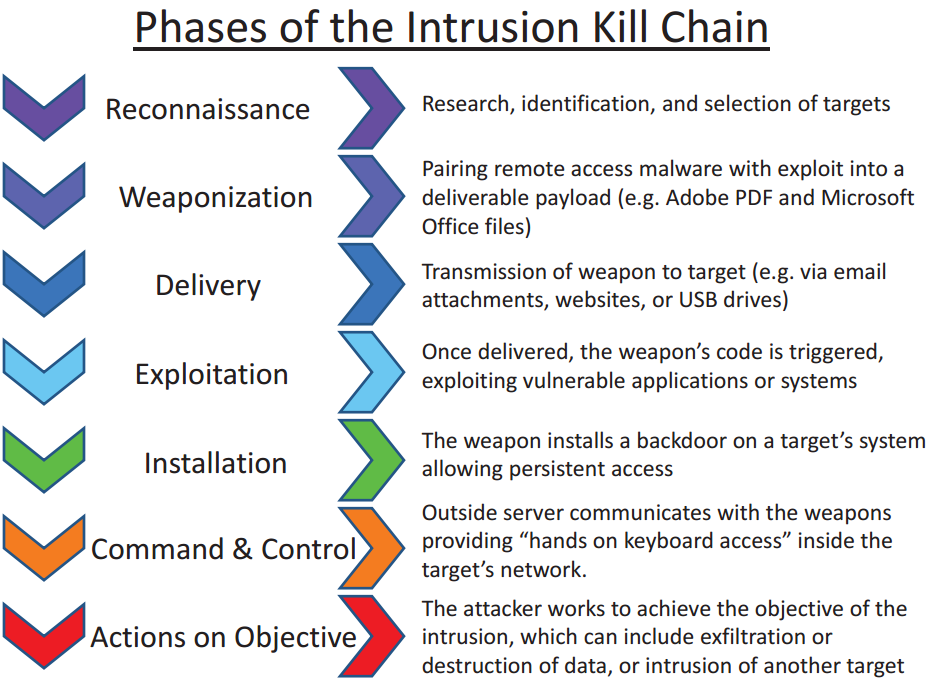
##### 

### Glossary

**Virtual Machine:** “... a virtual machine (VM) is an emulation of a computer system. Virtual machines are based on computer architectures and provide the functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination.”[[8]](#footnote-7)

**UAT:** “User acceptance testing (UAT) consists of a process of verifying that a solution works for the user. It is not system testing (ensuring software does not crash and meets documented requirements), but rather ensures that the solution will work for the user (i.e., tests that the user accepts the solution); software vendors often refer to this as "Beta testing".”[[9]](#footnote-8)

**Kill Chain:** “The term kill chain was originally used as a military concept related to the structure of an attack; consisting of target identification, force dispatch to target, decision and order to attack the target, and finally the destruction of the target. Conversely, the idea of "breaking" an opponent's kill chain is a method of defense or preemptive action. More recently, Lockheed Martin adapted this concept to information security, using it as a method for modeling intrusions on a computer network. The cyber kill chain model has seen some adoption in the information security community.”[[10]](#footnote-9)



**Open Source:** “The open-source model is a decentralized software development model that encourages open collaboration. The main principle of open-source software development is peer production, with products such as source code, blueprints, and documentation freely available to the public.” [[11]](#footnote-10)

**I/O:** “In computing, input/output or I/O (or, informally, io or IO) is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system. Inputs are the signals or data received by the system and outputs are the signals or data sent from it. The term can also be used as part of an action; to "perform I/O" is to perform an input or output operation.”[[12]](#footnote-11)

##### 

##### **Note**

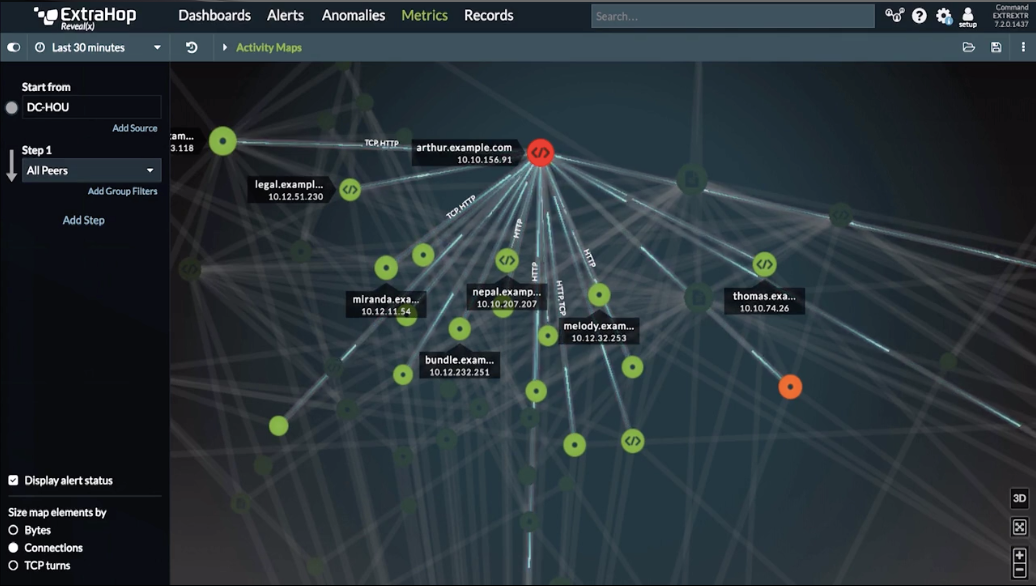
###### Vendor Recommendation

[**ExtraHop**](https://en.wikipedia.org/wiki/ExtraHop_Networks), one of my favorite SecOps Vendors has an amazing dashboard that allows SecOp Engineers to see threat intelligence from several different perspectives with a few clicks. As a NASA employee, I was able to demo the software and see the way their dashboard helps organize the flood of information coming from the myriad of devices on the network. It’s no wonder SecOps Engineers are overwhelmed, without the proper tools it is literally impossible to make sense of the activity happening within an organization's network.

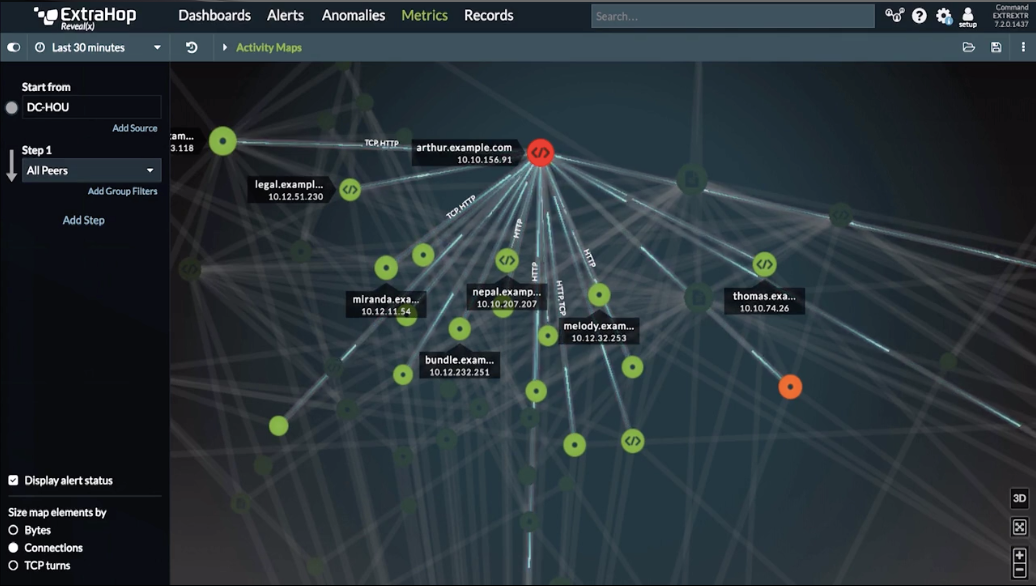
*“The core of ExtraHop technology is a passive network appliance that uses a network tap or port mirroring to receive network traffic, and then performs real-time full-stream reassembly to extract application-level protocol metrics and other custom-specified information contained in the transaction payload. A subset of these metrics are sent to the cloud where they are used as machine learning features to detect anomalous behavior that could indicate a data breach, for example.”* [[13]](#footnote-12) [[14]](#footnote-13)

Almost like the Alexa device sitting in our kitchen, ExtraHop is always listening, trying to understand if what it hears is normal or *anomalous.* This type of constant listening is pretty useless without tools that will make sense of the data. Threat Modeling and Classification is an important part of the ExtraHop value proposition, but it may be hard to believe. However, knowing that threats are never one dimensional helps software like ExtraHop detect threads of activity that may represent malicious behavior.

In the case of the[**breach at the DNC**](https://en.wikipedia.org/wiki/Democratic_National_Committee_cyber_attacks), there were several clues discovered in hindsight that would have tripped alarms: logins from strange locations within impossible timeframes, the elevation of user privileges followed by large movements of data. Yes, malware unlocked the door but breaking the lock alone was not the point of the breach. Network analysis, like the type offered by ExtraHop, could have stopped the hackers from getting the goods.



The screen above shows a visual network mapping of an event and how it has propagated through a network.



This screen shows aggregated data analysis that classified a chain of events as an attack.

#### 

#### Section 2

##### **Rationale**

Web Browsers are a popular surface for attacks because they may provide an open door to the World Wide Web, most can execute code in a powerful *Virtual Machine* and yet their regular use tends to lull users into a relaxed state, one in which they lower their guard. Even the most vigilant user can miss most of the *I/O* happening in the browser’s ecosystem, ie. cookies, cache, files sitting the downloads folder and more. Most of the software that protects endpoints does so by trying to intercept URLs but cannot protect against code that executes in a browser session once the user is already on a site.

Safe Browsing means more than just avoiding malicious websites. Threat actors can attack the browser from many different vectors. Security in the browser involves thinking about all the areas where user data is held and how this data is managed, understanding how scripts run while visiting a site, understanding how a site verifies its authenticity and managing the user's ability to install plugins to the browser or use the browser to run outside applications. **The** **Diamond Intrusion Model** [[15]](#footnote-14) describes how to understand threats beyond two-dimensional ideas that only look at an intruder and a victim. Instead, the author challenges us to understand the entire kill chain and how threats develop.

*“In its simplest form (Figure 1), the model describes that an adversary deploys a capability over some infrastructure against a victim. These activities are called events and are the atomic features. Analysts or machines populate the model’s vertices as events are discovered and detected. The vertices are linked with edges highlighting the natural relationship between the features. By pivoting across edges and within vertices, analysts expose more information about adversary operations and discover new capabilities, infrastructure, and victims.”*

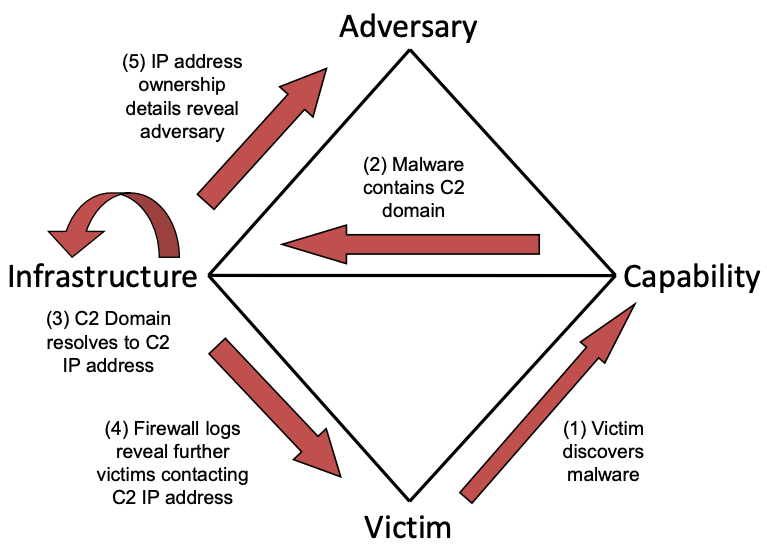
**

Figure 1

If we accept the assertions of the Diamond Model it is easy to understand why traditional methods of browser security are inadequate. Adversaries are not the malware we find in our networks or the vulnerabilities we uncover in software. **Adversaries are people** who develop capabilities. Whether these capabilities include exploitation of a software flaw is not prima facie. Sometimes exploiting software is not necessary is the user can we seduced into using the software in the wrong ways. Quoting the Diamond Model again:

*“...an adversary does not operate in a single event against a victim, but rather in a chain of causal events within a set of ordered phases in which, generally, each phase must be executed successfully to achieve their intent.”*

Overcoming traditional anti-virus or finding a hole in the firewall is akin to jumping a fence and finding an unlocked window. The breach event is not the objective, and it was not achieved without understanding the environment. By delivering and positioning AI at the frontline, inside the Browser, we effectively empower the “unlocked window” to **lock itself** because it recognizes that the family is asleep every evening by 10 pm, the smartphones of the owners that are usually always in their pockets are on their chargers ( not outside of the house ) and nobody has ever approached the window from the outside at 2 am in the history of recorded events. This smart decision making pushed to the edges of the network represent the future of security not just for Browsers but for IoT devices as well. Pushing this smart decision making into the engine that runs the Browser is a good first step.

Works Cited

Caltagirone, Sergio, et al. “The Diamond Model of Intrusion Analysis.” 2013, www.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf.

GoogleDevelopers. “Chromium's Multi-Process Architecture.” *YouTube*, YouTube, 29 Apr. 2009, www.youtube.com/watch?v=A0Z0ybTCHKs.

Dix, John. “ExtraHop Mines the Network to Glean Operations Intelligence.” *Network World*, Network World, 4 Oct. 2013, www.networkworld.com/article/2170569/lan-wan/extrahop-mines-the-network-to-glean-operations-intelligence.html.

“Endpoint Security.” *Wikipedia*, Wikimedia Foundation, 4 Oct. 2018, en.wikipedia.org/wiki/Endpoint\_security#Corporate\_network\_security.

Goodrich, Michael T., and Roberto Tamassia. *Introduction to Computer Security*. Pearson, 2014.

“How It Works.” *ExtraHop's IT Analytics Secret Sauce: How It's Made | ExtraHop*, ExtraHop, www.extrahop.com/products/how-it-works/.

Kassner, Michael. “Cyber Threat Hunting: How This Vulnerability Detection Strategy Gives Analysts an Edge.” *TechRepublic*, TechRepublic, 29 Apr. 2016, www.techrepublic.com/article/cyber-threat-hunting-why-this-active-strategy-gives-analysts-an-edge/.

“Multi-Process Architecture.” *The Chromium Projects*, Google, www.chromium.org/developers/design-documents/multi-process-architecture.

“The Broker Process.” *Sandbox*, Google, chromium.googlesource.com/chromium/src/ /master/docs/design/sandbox.md#the-broker-process.

“The Target Process.” *Sandbox*, Google, chromium.googlesource.com/chromium/src/ /master/docs/design/sandbox.md#the-target-process.

“Virtual Machine.” *Wikipedia*, Wikimedia Foundation, 16 Oct. 2018, en.wikipedia.org/wiki/Virtual\_machine.

**EMCS2000: Introduction to Computer Security:** Module 5 Report

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

### Section 1

#### Scenario

Access permission schemes pre-date computers, come in many flavors and can be very complex. With all of their complexity, most of them are still just asking a very basic question to determine if a subject has the right to access an object: ***“Who are you?”*** We may be able to reduce the overhead of access systems and vulnerabilities by exploring examples of real-world trust, and using machine learning to ask smarter questions at the right moments. One of my favorite lines from the movie “V for Vendetta” is when Evey asks V: *“Who are you?”* and he responds with a memorable answer:

**V**: *Who? Who is but the form following the function of what and what I am is a man in a mask.***Evey**: Well I can see that.  
**V**: *Of course you can, I’m not questioning your powers of observation, I’m merely remarking upon the paradox of asking a masked man* ***who*** *he is.*  
**Evey**: Oh, right. [[16]](#footnote-15)

**Who** someone is, often follows the function of **what** they are: doing, responsible for, overseeing, etc. RBAC tries to answer the question of who someone is with roles like admin, user, superuser, etc. However, I am not sure these type of enigmatic brief descriptions fully unravel the riddle of the “form following a function”, as V suggests. A manager, an admin, or a user are titles attached to no specific function. A User of what? An Admin of what? A Manager of what? We can provide a better answer when the system is programmed to ask a better question, the targets of security are curated to provide detail about appropriate usage and the titles associated with security classes acknowledge multifaceted relationships rather than two-dimensional permissions and privileges.

#### Security Objectives

The objective of this plan is to define access in ways that allow security policy to be defined from the deep curation of content in objects and their real-time relation to the subjects trying to access them. By automating curation and factoring relationship, we can create a policy that manages usage rather than just monitoring access based on an inherited policy title. We can also detect suspicious-looking access events, even if the user is authenticated, based on the acceptable range of historical usage.

#### Scope

##### General Target

The target for the plan will be the Active Directory FIle System, otherwise, know at ADFS. This system existing at NASA inside of SharePoint.

##### General Audience

The target of this plan will include all users of the active directory file system including the administrators and super administrators.

##### Affected Systems

The Permissions Policy Module in ADFS will be the specific target for this plan. This module controls the way the system allocates permission and is linked to SAML authentication systems that use smart cards and pins to authenticate users.

##### Development Stakeholders

System Engineers and System Administrators are the development stakeholders. System Administrators from various teams define the rules as the business leaders, and the System Engineers develop the solution to enforce the rules and monitor the system.

##### **Technical Description**

##### High-Level Abstract Concepts

We have many different ways of calculating trust and safety outside of computers. In the real world, we rely on relationships to measure trust. We know that a title alone does not entitle someone to access. For example, a child’s mother has authority over them as opposed to just any random person with the title mother. The title “mother” alone does not grant access. However, a mother may pass authority to another family member or even a stranger, who will consequently have authority over the child, even though they don’t have the title. This type of permission grant is nuanced, but we can distill the transfer of permission to a few key points. The subject ( the mother ) is the creator of the object ( the child ) and therefore inherits the natural authority to grant permission. The mother also has a relationship with the child, a history of interactions. This is relevant because a mother with no history of caring for their child would not hold the same authority ( in the case the mother had given the child up for adoption ). Lastly, we can assume the mother has some interest in the security of the child, so the mother is checking any person that she grants permission to for some level of trust. To summarize, we need to check three things when granting permission, hierarchy ( is the subject a parent to the object? ), relationship ( is the subject connected to the object? ) and trust ( can the subject trust the grantee of the permissions with the object, and what is the range of that trust ).

There are no actual children in a file system, but let’s examine the analogy and see if the ideas of hierarchy, relationship, and trust hold true. A user creates a file ( the user is the parent of the file ), the user frequently updates the file ( a history of interactions ) and then the user may assign some importance to the file ( shows a concern for the file’s security ). That last step rarely happens. We have too many “children” in the digital space and rarely make time to assign a level of importance to each document, picture or file we make. And it follows since there is rarely detailed categorization of security for each document, file, and picture, they all default to the same level of security, which is usually a function of referencing predefined access list. Furthermore, it is rare that a user who is granted permission to a file has time to go through the same level of vetting that a mother would conduct with a potential babysitter. So, we can conclude that habits surrounding file creation and the time associated with proper vetting may prevent us from treating our files with the same level of care a mother would give their children. But, what if the process of categorization and vetting could be automated with technology?

##### Challenging Some Older Axioms

RBAC, one of the most popular access control systems, has a notion of hierarchy, and maybe some notion of trust, but nothing that defines a relationship between the subject and object. As a matter of fact one of the axioms purported in the related LBAC model says that security “flows” from one object to another, however, I feel this definition is a bit antiquated. Sandhu says “Information flow policies are concerned with the *flow* of information from one security class to another. In a system, information actually flows from one object to another." [[17]](#footnote-16) In modern information systems I don’t see a lot of information flowing from one object to another, but I do see data object being exposed in many different contexts, sometimes simultaneously. I am going to challenge this axiom and say that security doesn’t “flow” from one object to another. Instead, information is exposed in many contexts and the security policy should transform based on usage, context and of course the viewer. The owner, creator or grantor grants a range of permission either explicitly or implicitly and expects this policy to be enforced in context. The grant permission depends on context, relationship with the grantee and includes a range of security classes that have upper and lower bounds but contains infinite inner classes that depend on usage. That last part also challenges the first of Dennings axioms, that the set of security classes are finite.[[18]](#footnote-17) This understanding of information does factor in the transformation of data and the many ways it can be reposted, reused and transformed in a way that aligns with or violates the intent of the security classes.

Returning to the example of a file created by a user that they update on a regular basis. If they assign a moderate security classification to this file it could mean several things beyond who is able to access it. Where the file is stored, the number of times the file can be accessed, the ways the file can be accessed and how the data in the file is used may also be a part of the security policy. In fact usage of a file is exponential more complex than access, because security around usage asks “what are you going to do with this?” The fact that access controls fail to ask this question coincides with the recent uproar about data privacy and commercial exploitation of private data.

##### Creating a Curated System

Social Media has trained most of us to carefully curate our posts by allowing us to choose if a post is public or private. In some cases, we can also choose specific people the post can be seen by or is hidden from. What if these types of controls existed for every file in the file system so that when users created files they are encouraged to assign or accept a security classification? Google recently released a feature in Gmail that will finish your sentences. How much harder would it be to create a machine learning agent to suggest a security classification for a document along with a list of recommended allowed users. The curation must go much deeper. The curation should also determine if the document contains Personally Identifying Information or other sensitive information. Maybe in the future its ok that users have a document on their computer called passwords because the automated curation would encrypt this file for everyone else even someone logged in as root? The point is that we can take the work out of curation by letting an algorithm do the work.

##### Trust Factoring in Relationships

Imagine again our mother-child example, if a stranger wants to exert authority over the child this is possible but only with the mother’s explicit permission. The mother first establishes the relationship with the stranger, who then becomes the babysitter for example, and as a part of her vetting process, she grants limited authority to the babysitter. The babysitter never becomes the mother or replaces the mother, he or she acts on the authority given by the mother. The mother grants permission, but with some explicit or implicit usage policy.

**Now, let’s think of an attack in a common kill chain.** After conducting reconnaissance an attacker finds a vulnerability in the RBAC model, one that allows them to use a role with low privileges to make a decision that should be reserved for a higher authority, or they find a way to elevate the privileges of a user without authorization from the super admin. Translating this back into our analogy, this is the equivalent of a stranger sneaking into the house and asserting “the authority” of the mother, or a person disguising themselves as the mother in an attempt to fool the child. What makes our analogy different than RBAC is the fact there is only one legitimate mother. In an RBAC system, there can be many admins, in large organizations admins are created every day. In fact, there may be admins ( new and old ) without a relationship to the data they seek to view or control. How do we solve this problem?

**Going back to our analogy let’s think deeply about what the mother’s grant of permission means.**  If the babysitter is watching the child at the mother’s house, when she grants permission to watch the child she also grants permission to enter the house, walk on the floor, sit on the couch. The babysitter doesn’t have to ask for permission each and every breath. This is not to say there is not an upper range for the permission granted. The babysitter is probably usually not permitted to go in the parent's bedroom, or take the fancy silverware and stuff it in their backpack. Furthermore, in the physical world, a grant of permission comes with rules of usage based on relationship. If the mother’s own mother watches the child the grant of permission is different than if a 14-year-old neighbor or a friend of a friend’s daughter watches the same child. All of these ideas about permission and access make sense in the physical world, but do they translate when thinking about a file system?

##### Access Calculations

Role-Based Access Controls, Attribute Based Access Controls, and Lattice-Based Access Controls all give us some notion of control over an aspect of systems that users interact with.

In **RBAC** hierarchy is represented by the sets U, R, P, S (users, roles, permissions, and sessions, respectively) and defined by these base formulas.  
• UA ⊆ U × R (user assignment)  
• PA ⊆ R × P (permission assignment)  
• RH ⊆ R × R is a partial order called the role hierarchy or role dominance relation written as ≤.[[19]](#footnote-18)

**ABAC** is very similar but switches the concept of roles with attributes.

**LBAC** popular among engineering who create REST Services and API’s describes security “in terms of the lattice (a partial order set) where each object and subject have a greatest lower bound (meet) and least upper bound (join) of access rights.” [[20]](#footnote-19)

Knowing this, how do the traditional models calculate access and usage? For example, two nuclear scientists are working on some of the same projects, with the same level of security clearance, for the same organization. However, they may or may not be permitted to see the same files in a given directory if some of the documents include discussions with military intelligence about HEARTBEAT, for example. Even among people with the same level of security clearance, information about certain topics may be privileged on a need to know basis. Using traditional RBAC having files with varying access levels in the same folder would create a complex security policy conundrum, one where a special policy is attached to the document or a restriction where a document cannot be included in a folder with other documents. Using ABAC to filter the documents would mean creating and assigning some special attributes only accessible to one of the scientist. With highly sensitive documents this still might work, but as we mentioned previously people rarely take the time to describe access and usage at a granular level, per document. Using LBAC to filter the documents would mean assigning the document to a security class within certain bounds for one scientist and out of the bounds for the other, a class that might not exist outside of blocking information about one topic.

**By using machine learning to perform curation on data objects** which feeds a dynamic policy engine, a document with sensitive conversations between one nuclear scientist and military intelligence about HEARTBEAT, can be hidden from the other nuclear scientist based on the content in the document not on a policy attached to a folder, a predetermined uncommon attribute or a complex set of security bounds. The curation engine recognizes the conversation as privileged, prompts the user for guidance, and feeds the curation into a policy engine that determines access based on the relation of the subject to the document. The rule created in the engine is not transitive to any nuclear scientist with the same clearance, but only those that have a relationship with the topic and a history in the conversation. In a sense the control access threats each piece of content like a sensitive social media post, allowing the user to fine-tune the audience and usage, prompting them with curation ques based on similar bits of content. Here is a sample interaction from an email implementation of this new model:

( User is sending an email with sensitive information about HEARTBEAT to several recipients )

**Email Client:** “It looks like this email contains a link to confidential information about a program you have access to called: HEARTBEAT. Are you sure you want to include John Shirley, Richie Valentine, and Lionel Richie? There is no record they have authorization”

( User clicks an option to ADD John Shirley, and redact the mentions of HEARTBEAT for the other users )

**Email**  “Sent a request to add John Shirley to Patrick Stewart, information will be redacted until access it authorized”

This is not just good for filtering documents in a folder, but it can also be used to prevent unauthorized access. The system can restrict access when the user or creator behaves in a strange way or seems likely to be an imposter since the automated curation also includes a range of acceptable/normal access patterns. Again, creating this is RBAC would be next to impossible, but machine learning can render an analysis when the assertion of access or authority looks strange. When does the assertion of authority look strange? When a new subject, with a new status, asserts authority over an object it has never had a relationship with before. But not only that :

Asserting authority over an object after entering from a strange/unknown location

*( the Mom climbing in the window )*

Asserting a type of authority that has ultimate consequences for the object

*( the Mom tries to kill the child )*

*Giving authority to a new subject without consensus*

*( the Mom giving a child away to a stranger )*

**And notice what all of these examples have in common.**  *They all involve a risk to the object aka the child.* If we ask a smart question about the trust the child object requires we might be able to understand the necessary controls required for access and what that means.

##### Scoring Trust by First Measuring Distrust

I want to propose that we measure trust inversely by measuring distrust. If distrust is 0 than the subject is completely trusted. If distrust is 100 then the subject is completely untrusted. With a lower distrust score, the subject is granted more freedom of usage. However, I would maintain that a distrust score of 0, still should be calculated in real time base on activity and usage. A user signing in for the first time from a Russian IP address at 1 am Pacific Standard Time on a Sunday my now violate the login access policy, but may not have enough trust to view documents with highly classified information that requires 0 distrust. This model allows the system to make intelligent decisions about access in real time without forcing all or nothing scenarios.

##### Creating Nuanced Layers of Access

**Each level of access involves many different layers.** Before you can read something you have to be able to see it. And usually seeing it involves the ability to look at its details. Seeing something and looking at details may seem similar but there are some nuances. Let’s exchange some terms. Is the object visible? Can you see the details? Visibility without details could let us know there is *something* there, without knowing what it is. Visibility with the ability to see details means knowing it is there and being able to see details like file name, file size, file type, etc. A subject could have visibility and details and still not be able to read. The same levels of access exist when trying to discern whether a subject copy an object or move it or destroy it.

Furthermore, it's understood that if a subject has the power to destroy an object the subject has the ultimate authority over the object. I would argue just like in our analogy, where even the mother does not have the right to destroy the child, the power to destroy objects should not exist. It follows that every parent, owner or creator of an object should not necessarily have the power to destroy it. Taking the analogy further, one could argue that God has the legitimate power to destroy everything, anything, and anyone. ( Not to start a religious debate, just using this as an analogy ). Let God in this analogy represent the **root user**. I am sure several Sys Admins would agree that while administrators may invoke “the power of God,” no admin should be “acting like God” all the time. This is the reason we have **sudo** in bash and use it sparingly. Defining when an admin has the right to “invoke God” is an important discussion, but not one we can have fully here. We can agree however that extended use of this God authority by any subject in the system represents a threat to the system and all of the objects.

Again notice the commonality in the details discussed in the previous paragraph. They involve an evaluation of the risk to the object ( the child ).

##### Implementation

At NASA permission scheme/process are pretty antiquated. Gaining access to a resource, whether that resource is computational, physical or access related, involves submitting a formal request that must be checked by a sponsor and a supervisor. The request may also involve required training, security clearances and even US citizenship.

I am going to call what I have described up until this point a **Dynamic Relational Trust-Based Access Control ( DRTBAC ).**  This keeps track of the content that is produced by the user and aids in the curation of content for the use in a real-time policy engine. Much like the AI policy engine in the Chromium project this engine uses the analyzes the content and tries to come up with a policy based on the sensitivity of the content. In many ways, the NASA model has the same controls, but DRTBAC model automates the process and allows for the possibility of security within documents.

Implementation of DRTBAC involves installation of a custom agent in NASA HostScan system, SAML integration with Sharepoint for the PIV authentication methods and robust backend to train the machine learning model and retain policy data.

##### Assigning Roles

Assigning roles in DRTBAC is much different than previous models. DRTBAC is more interested in the projects and groups the user associated with instead of their hierarchy or title. Once a user is assigned to a group they relative access to documents with users that share their security clearance and work descriptions. Continuing with the pair of Nuclear Scientist, once Richard Sherman gets his NASA ID and laptop he still doesn’t have access to anything outside of his machine. To get access he must be associated with a group working on a project and as user start interacting with Richard and sharing content access is granted by the owner of the content. In some ways, this mimics the Discretionary Access Control Model ( the model currently used at NASA ), with the key difference that the system is curating and monitoring the way Richard interacts with co-workers and is generating suggestions for access as he works, verifying this access with the owners of the system against a record of his security clearance.

##### Troubleshooting

Troubleshooting the DRTBAC system may involve years of development and testing. Creating an ML model that is able to properly classify sensitive information will evolve significant effort and a lot of testing. The primary challenge to ensure that the NLP function of the model is working and the transfer of data between the authorization agents and the user systems is seamless.

##### Exceptions

Systems that are not connected to the internal network will not subject to this new system.

### Section 2

#### Rationale

The rationale and common sense justification for this new access model have been discussed throughout the plan, but to summarize benefits, it is the only model that allows granular control without creating enormous overhead or relying on overclassification of subjects and objects. We live in a time where copy and paste might be as dangerous as a breach from an attacker. With granular controls, a word document, for example, can disallow copy and paste for a block of text for one user and allow it for another. This type of control is just not possible in other models without a lot of administration definitions.

Furthermore, RBAC is the most attacked access model according to threat reporting organizations. Access Models based on roles are well-known targets for attackers. Changing a role, circumventing a role, or hacking the authentication model is the type of work hackers are familiar with. Fighting policies created by AI is not such a well-known problem. In the DRTBAC Model merely stealing credentials doesn’t ensure access and usage of the file system’s most sensitive materials. An attacker with a user’s credentials would also need to mimic usage in the range of the user. This makes data exfiltration, elevating privileges and installing malware high problematic. The attacker is essential trying to match a usage fingerprint that they can’t see or analyze.

#### Note

Designing systems based on Data Science is extremely hard. Embarking on a project like this is very ambitious and will probably fail several times before it produces any results. The key making Data Science algorithms work well is knowing when to stop and start over. Also getting too many positive results early on is usually a sign of a poorly written algorithm or sample data that doesn’t have enough variation.

My criticism of previous models should not be taken as negative. Computer Science has gone through some groundbreaking changes just in the last 7 years. Many of the access models I discussed were written in the 1970’s and may be updated in the late 1990’s before the true advent of serverless systems and cloud systems. In the era of the cloud, SysAdmins may not know the exact location of data, so to say that it “flows” from one place to another is not very meaningful. Much of the most sensitive data users have resides in their email, a point John Podesta and the DNC know all too well. The idea of securing data in the same way we secure a stack of cash sitting in a vault doesn’t make much sense. Imagine someone taking that cash and throwing it in the air in the middle of Times Square, how do you secure it now? We are moving towards a time when data itself has to be smart enough to secure itself.

#### 

#### Glossary

**bash**: Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. First released in 1989, it has been distributed widely as the default login shell for most Linux distributions and Apple's macOS (formerly OS X). A version is also available for Windows 10. It is also the default user shell in Solaris 11.[[21]](#footnote-20)

**HEARTBEAT**: HeartBeat was an alleged NSA program that utilized resources in another clandestine program called PRISM. Edward Snowden claimed this program allowed the NSA to keep track of the “heartbeat” of every device connected to the network.

**Root User**: In computing, the superuser is a special user account used for system administration. Depending on the operating system (OS), the actual name of this account might be root, administrator, admin or supervisor. In some cases, the actual name of the account is not the determining factor; on Unix-like systems, for example, the user with a user identifier (UID) of zero is the superuser, regardless of the name of that account; and in systems which implement a role-based security model, any user with the role of superuser (or its synonyms) can carry out all actions of the superuser account. The principle of least privilege recommends that most users and applications run under an ordinary account to perform their work, as a superuser account is capable of making unrestricted, potentially adverse, system-wide changes. [[22]](#footnote-21)

1. Goodrich, Michael T., and Roberto Tamassia. *Introduction to Computer Security*. Pearson, 2014. [↑](#footnote-ref-0)
2. “Endpoint Security.” *Wikipedia*, Wikimedia Foundation, 4 Oct. 2018, en.wikipedia.org/wiki/Endpoint\_security#Corporate\_network\_security. [↑](#footnote-ref-1)
3. GoogleDevelopers. “Chromium's Multi-Process Architecture.” *YouTube*, YouTube, 29 Apr. 2009, www.youtube.com/watch?v=A0Z0ybTCHKs. [↑](#footnote-ref-2)
4. “Multi-Process Architecture.” *The Chromium Projects*, Google, www.chromium.org/developers/design-documents/multi-process-architecture. [↑](#footnote-ref-3)
5. “The Target Process.” *Sandbox*, Google, chromium.googlesource.com/chromium/src/ /master/docs/design/sandbox.md#the-target-process. [↑](#footnote-ref-4)
6. Davis, Brian. “Post-Work: Discussion - Recovering After a Password Breach!” *Post-Work: Discussion - Recovering After a Password Breach!*, Brian Russel Davis, 8 Oct. 2018, canvas.brown.edu/courses/1076326/discussion\_topics/3564539. [↑](#footnote-ref-5)
7. Kassner, Michael. “Cyber Threat Hunting: How This Vulnerability Detection Strategy Gives Analysts an Edge.” *TechRepublic*, TechRepublic, 29 Apr. 2016, www.techrepublic.com/article/cyber-threat-hunting-why-this-active-strategy-gives-analysts-an-edge/. [↑](#footnote-ref-6)
8. “Virtual Machine.” *Wikipedia*, Wikimedia Foundation, 16 Oct. 2018, en.wikipedia.org/wiki/Virtual\_machine. [↑](#footnote-ref-7)
9. Cimperman, Rob (2006). *UAT Defined: A Guide to Practical User Acceptance Testing*. Pearson Education. pp. Chapter 2. ISBN 9780132702621 [↑](#footnote-ref-8)
10. "Kill Chain Approach". Chief of Naval Operations. April 23, 2013. Archived from the original on June 13, 2013. [↑](#footnote-ref-9)
11. Levine, S. S., & Prietula, M. J. (2014). Open collaboration for innovation: Principles and performance. Organization Science, 25(5), 1414-1433. doi: 10.1287/orsc.2013.0872 [↑](#footnote-ref-10)
12. “Input/Output.” *Wikipedia*, Wikimedia Foundation, 10 Feb. 2018, en.wikipedia.org/wiki/Input/output. [↑](#footnote-ref-11)
13. “How It Works.” *ExtraHop's IT Analytics Secret Sauce: How It's Made | ExtraHop*, ExtraHop, www.extrahop.com/products/how-it-works/. [↑](#footnote-ref-12)
14. Dix, John. “ExtraHop Mines the Network to Glean Operations Intelligence.” *Network World*, Network World, 4 Oct. 2013, www.networkworld.com/article/2170569/lan-wan/extrahop-mines-the-network-to-glean-operations-intelligence.html. [↑](#footnote-ref-13)
15. Caltagirone, Sergio, et al. “The Diamond Model of Intrusion Analysis.” 2013, www.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf. [↑](#footnote-ref-14)
16. McTeigue, James, director. *V For Vendetta*. Warner Bros., 2006. [↑](#footnote-ref-15)
17. Sandhu, Ravi S. (1993). "Lattice-based access control models" (PDF). IEEE Computer. 26 (11): 9–19. doi:10.1109/2.241422 [↑](#footnote-ref-16)
18. Denning, Dorothy E. (1976). "A lattice model of secure information flow" (PDF). Communications of the ACM. 19 (5): 236–243. doi:10.1145/360051.360056 [↑](#footnote-ref-17)
19. Sohr, Karsten & Drouineaud, Michael & Ahn, G.-J & Gogolla, Martin. (2008). Analyzing and Managing Role-Based Access Control Policies. Knowledge and Data Engineering, IEEE Transactions on. 20. 924-939. 10.1109/TKDE.2008.28. [↑](#footnote-ref-18)
20. “Lattice-Based Access Control.” *Wikipedia*, Wikimedia Foundation, 14 May 2018, en.wikipedia.org/wiki/Lattice-based\_access\_control. [↑](#footnote-ref-19)
21. Hamilton, Naomi (May 30, 2008), "The A-Z of Programming Languages: BASH/Bourne-Again Shell", Computerworld: 2, retrieved March 21, 2011, [↑](#footnote-ref-20)
22. "getpwuid". opengroup.org. Retrieved 16 September 2015. [↑](#footnote-ref-21)