Command Shift: Exploring Modern Gaming Technologies to Create Next-Generation OCO Interfaces

Chad Caison
KEYW Corporation
Hanover, MD
ccaison@keywcorp.com

Jennifer McNamara
BreakAway Games
Hunt Valley, MD
jmcnamara@breakawayltd.com

ABSTRACT

Offensive Cyber Operations (OCO) are complex tasks involving abstract and logical concepts that are difficult for end users to synthesize and engage with. Typical user engagement involves using a keyboard and mouse, and working with Command Line or Graphical User Interfaces (CLIs and GUIs). While these tools let users command tools, they do not provide robust situational awareness and they require extensive training and experience to achieve competence. This paper describes the analysis, design, and development of a new cyber interface that uses input and visualization methods borrowed from the game industry to fully immerse the operator more naturally in the cyber battlespace. This type of tool has never been used in the OCO space and has neither been openly welcomed nor understood within the community. Until now, stakeholders have not seen a prototype to demonstrate the potential of this more natural type of interface, which gives the operator a better understanding of the abstract environment to facilitate better decision making and reduce human mistakes. To create this new user interface, two organizations collaborated: one representing current OCO training and state-of-the-art OCO tools, and the other representing game design. Together, these organizations designed a new cyber interface focused on three primary goals: to reduce complexity and training time, to improve situational awareness, and to reduce human error. This paper discusses the standard OCO work environment and its challenges for end users, the results of our user analysis that drove the design process, the game-based hardware and software considerations used by the team, and the prototype interface itself, along with informal playtest reactions from end users.

ABOUT THE AUTHORS

Chad Caison is the Technical Director for KEYW Corporation's Advance Cyber Operations Sector. He provides technical guidance for the sector's train, man, and equip capabilities in support of computer network operations. Mr. Caison is responsible for the technical vision of KEYW's offensive and defensive cyber operations capabilities and services, and has authored several commercial and government-centric offensive security courses. Since 2008, KEYW Corporation has been a provider of OCO and Defensive Cyber Operations (DCO) training for the Department of Defense (DoD), and has trained over 2,000 students. Mr. Caison served eight years in the US Army: four years with the infantry and four as a military intelligence officer. He has served in Iraq at the Combined Intelligence Operations Center. Mr. Caison also served at Fort Meade, Maryland, at the National Security Agency as a Global Network Analyst. Mr. Caison graduated from North Carolina State University with a Bachelor's Degree in Sociology. He graduated with a Master's Degree in Information Systems from the University of Maryland, Baltimore County. Mr. Caison has spoken at Johns Hopkins Applied Physics Laboratory Conferences in Information Visualization on advanced visualizations for offensive security and at the Computer and Enterprise Investigation Conference (CEIC) on malware persistence techniques. He has also served as a panelist at the Cyber Security Colloquium at George Mason University and spoken at the I/ITSEC 2014 Ignite presentations.

Jennifer McNamara is the Vice President of Serious Games at BreakAway, Ltd., creating serious games that serve the corporate, defense, homeland security and medical communities. Jenn holds a B.S. in Cognitive Psychology from Drexel University and a M.Ed. in Instructional Systems from The Pennsylvania State University. She strives to help professionalize and advance the state of the serious games industry by serving on the Serious Games Showcase and Challenge IPT.

Command Shift: Exploring Modern Gaming Technologies to Create Next-Generation OCO Interfaces

Chad Caison
KEYW Corporation
Hanover, MD
ccaison@keywcorp.com

Jennifer McNamara
BreakAway Games
Hunt Valley, MD
jmcnamara@breakawayltd.com

STANDARD OCO WORK ENVIRONMENT

Offensive Cyber Operations (OCO) are complex and require operators to maintain awareness of the target network, the state of the connection, complex command syntax, and tool workflows to be effective. For those unfamiliar with OCO, Joint Publication 3-12 (Joint Chiefs of Staff, 2013) provides an overview of OCO doctrine. OCO tools evolve from command line tools, and so end users must build mental models of target networks by synthesizing textual data and linking graphs. While most interactive OCO tools share a core set of functions, the quality of the interface that the operator uses to take actions can dramatically impact the operator's performance. Poorly designed interfaces can prolong training times, increase errors, and fail to provide operators with the situational awareness needed to be efficient and productive.

The tools used to gain access and explore a target typically have a command line interface (CLI) or a graphical user interface (GUI). CLI tools such as nmap (nmap, 2011) and Metasploit's msfconsole (Offensive Security, 2010) evolved from early tool suites that worked in Linux terminal environments. These tools were written in C/C++, Python, or Perl, which accept user input as text commands and provide outputs in the text.

The intrusion kill chain for cyber network attacks walks through the stages of: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and finally actions on objectives (Hutchins, Cloppert, & Amin, 2011). A simple workflow that puts the intrusion kill chain into action involves the following tasks: (1) identify computers to exploit, (2) analyze the data returned to select the correct exploit, (3) find the exploit, (4) load the exploit into the context of the tool, (5) set the exploit parameters, (6) run the exploit, and (7) perform actions on the remote host. The syntax to use nmap and Metasploit to accomplish these tasks is shown in Figure 1.

```
1. Identify computers to exploit, via nmap
Nmap -vv -PN -n -sT -p445 192.168.169.2-255

2. Analyze the data
Nmap scan report for 192.168.169.150
Host is up (0.0032s latency).
Scanned at 2014-01-11 05:19:27 PST for 23s
PORT    STATE SERVICE
445/tcp open  microsoft-ds

3. Find the exploit, via msfconsole
Search type:exploit platform:windows smb
       [Returns list of over 20 exploit candidates]


4. Load the exploit into context
use exploit/windows/smb/ms08_067_netapi

5. Set the  exploit parameters
set payload windows/meterpreter/reverse_tcp
set rhost 192.168.169.150
set lhost 192.168.169.140

6. Run the exploit
Exploit

7. Perform actions on the remote host. Meterpreter,
the backdoor left behind after exploitation, has
over 22 basic commands, and over 30 commands that
extract data from the target. The implant also comes
with extensions and scripts that add functionality.
```

Figure 1. Example command and workflow for exploiting a box

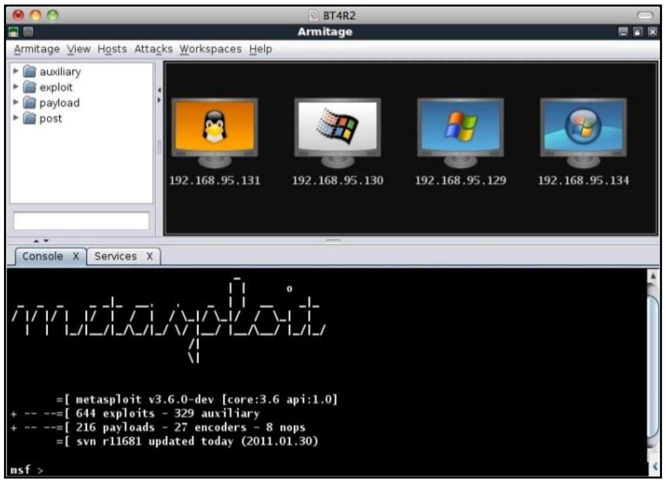
This seven-step workflow does not include the basic user/administration commands to install or launch the tools. This workflow is also very simplistic; in a sophisticated red team exercise, communications between attackers and targets are redirected through other victims to obfuscate the attackers' point of origin. This addition to the workflow would add several more complex commands and include tunneling concepts. This workflow also assumes that the operator is knowledgeable in the tool and the theory of remote exploitation, as it does not perform any error checking. If a wrong value is put into the parameters, then the tool may send the data to the wrong computer, which can jeopardize operational security and violate the Rules of Engagement.

Many popular tools have a GUI, which includes some form of text terminal to accept user inputs and render responses, and menu items that can be clicked using a mouse (see Figure 2).

More sophisticated GUI implementations attempt to visualize attacker-target relationships with a link graph (see Figure 3).

GUIs provide more guidance for invoking a tool's functionality than CLIs. For instance, they use input boxes and menu items with descriptive names, which help the user navigate the front-end. However, users still must understand the "buttonology". A GUI can still be time-intensive for training and operating, as users switch between the mouse and keyboard.

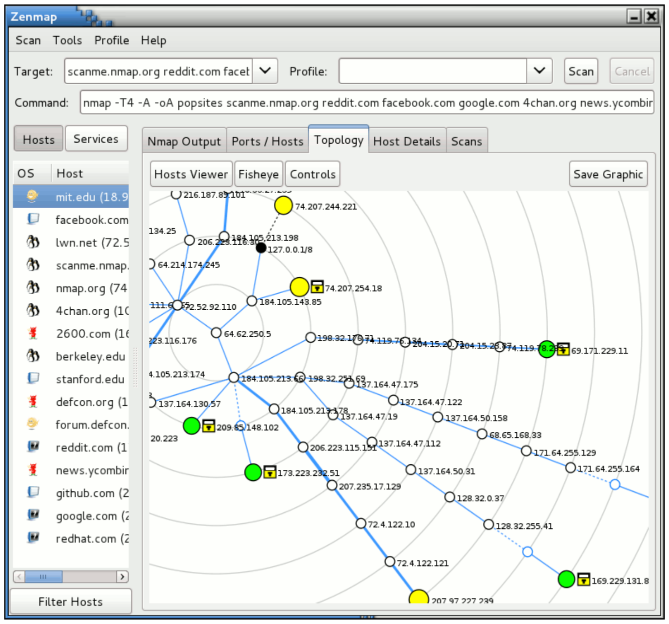
Both CLIs and GUIs can be difficult to learn as more complex actions are incorporated into the tools. This is compounded as human-computer interaction progresses away from traditional forms of input and towards more human-accessible inputs such as gesturing, tactile manipulation, and vocal commands. Over time, we anticipate that more (and younger) users will become less comfortable with the traditional OCO interfaces.



```
       =[ metasploit v3.6.0-dev [core:3.6 api:1.0]
+ -- --=[ 644 exploits - 329 auxiliary
+ -- --=[ 216 payloads - 27 encoders - 8 nops
       =[ svn r11681 updated today (2011.01.30)

msf >
```

Figure 2. Armitage, image from Linuxjournal.com



Target: scanme.nmap.org reddit.com facet

Profile:

Command: nmap -T4 -A -oA popsites scanme.nmap.org reddit.com facebook.com google.com 4chan.org news.ycombir

Figure 3. Zenmap, topology view

Challenges for End Users

With the advent of the cyber domain as a new domain of warfare, there has been an increased demand for trained DoD personnel. KEYW Corporation is a provider of OCO and Defensive Cyber Operations (DCO) training for the Department of Defense (DoD), and has trained over 2,000 students. Initially, personnel entering the training pipeline held technical undergrad and graduate degrees and had extensive operating system and network knowledge, as well as developer experience. As more recruits joined the organization, DoD personnel with less technical backgrounds have been asked to perform at a similar capacity for cyber operations, and as OCO and DCO trainers, we have observed students struggling with operating toolsets – these struggles motivated the development discussed in this paper.

Three factors that appear to impede a student's ability to learn and use a given tool have been observed: (1) command set complexity in the interface, (2) a lack of mission narrative to help the student gain situational awareness, and (3) a wide range of opportunities for user input to be prone to error. Thus, tool interfaces with excessive complexity, poor story-telling features, and no effort to prevent simple errors through thoughtful design are more difficult to learn and master.

Command Complexity

Many current operator tools have a CLI, a type of interface created before desktops had graphical interfaces and yet still used today to interact with servers and many administrative tools. Computer security culture often views CLI use as an indicator of technical competency, as users who can quickly type commands and mentally weave them together into a comprehensive workflow are viewed as more proficient than those who rely on pointing and clicking with a mouse.

However, the students that enter our training pipelines today, despite using computers since they were small children and having technology infused into every aspect of their daily lives, are largely unfamiliar with CLIs, and their difficulties learning to use CLIs are compounded by the tools' complex command structures and lack of workflows. For example, many CLIs enable base commands with flag options and arguments that are constructed to execute program functions, so users must have an advanced understanding of the tool's commands to effectively string commands together to accomplish complex tasks.

Human Error

Current operator tools do not account for human mistakes, and both CLI and GUI tools that accept user inputs can be compromised by user errors. So while the majority of professional tools do recognize syntax errors (e.g., when a number is replaced with a letter), they do not recognize errors associated with input values (e.g., when one number is replaced with another number).

For example, suppose an operator selects an exploit that they believe will succeed on the target. The operator must now input the target IP into the exploit. If the operator types the IP address in the proper format but with a wrong digit, perhaps off by one, then the packet containing the exploit is sent to the wrong computer, which may trigger the target's security systems. This type of error could be prevented if the tool's interface allowed users to quickly and easily check references and import data, such as correct IP addresses.

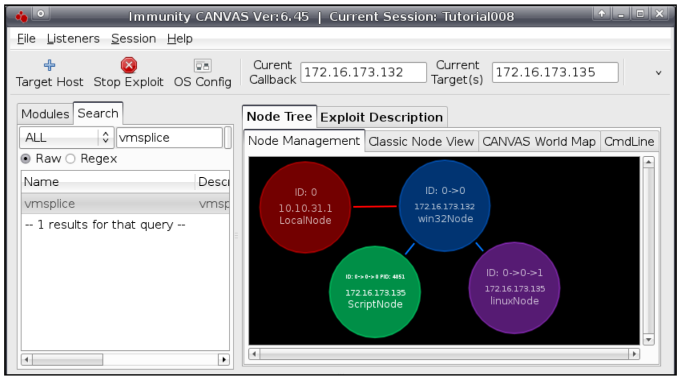
Another common human mistake is selecting an incorrect exploit. In the typical workflow, the operator analyzes target data and then identifies an appropriate exploit. If the operator has a suite of exploits tailored for specific platform architectures, operating systems, and service versions, then they must choose the correct variant in order to gain target access. If they select the wrong exploit (through ignorance, error, etc.), the impact can be tremendous, such as crashing the target and generating logs that reveal the operator to the target.

In addition to the errors discussed above, a mission can be affected by forces outside of the interface. The operator may be emotionally distraught, physically tired, or angry and frustrated, which can cloud their judgment and decision making, thus impacting mission performance. Therefore, a superior tool interface must be designed to account for these states and to help the operator to easily manipulate data and invoke tool functions.

Cyber Situational Awareness

A lack of useful tool feedback or data visualizations can make OCO increasingly difficult. Traditionally, penetration testing tools were CLI programs, but large command responses have become difficult to read and synthesize.

Over time, professional tools have attempted to provide more accessible interfaces. Some GUIs visualize the target network using a link-graph (see Figure 4), but this graph can suffer from an overly simplified view that lacks key data. Alternatively, link-graphs can become so complex that they take on a "fuzz ball" shape, making it difficult to see relationships and data-enriched nodes.



Curent Callback: 172.16.173.132
Current Target(s): 172.16.173.135
ALL | vmsplice
vmsplice | vmsp
-- 1 results for that query --

Figure 4. Immunity Canvas node tree

Most tools do not enrich the view with additional data feeds. Instead, the interface only accounts for the commands and data that it generates, and does not render additional data from other sources, which can preclude the operator from making better-informed decisions. For example, over long-term operations within a target network, an operator will need to incorporate historical actions, but if the interface does not include historical data, then the operator may recreate past mistakes or conduct similar actions that build a pattern that can be identified by the target and used as indicators of compromise. Once identified, the operator may be easily identified and quickly caught.

Poor data visualizations also result in poor situational awareness, lack of pre-attentive visualization techniques in the interface can create more costly search time for the user. Many interfaces fail to use the properties of color and position to help separate the dimensions of the data set, grouping categorical data and using hue/saturation to draw the operator's attention to what is most import.

Training Time

Organizations that maintain a large staff of skilled OCO tool operators are highly sensitive to training requirements. An individual with the appropriate prerequisite knowledge and experience may need weeks or even months of training before they can use a new tool on mission. In several of the courses we teach, a minimum of twenty percent of class time is dedicated to CLI training, and students with limited CLI experience need even more time to obtain a basic level of comfort. And when complex concepts such as tunneling—techniques to encrypt and route data between nodes—are introduced, the challenges of the tools often prevent students from grasping the underlying theory as they struggle to accomplish lesson tasks.

In our DoD-centric courses, there are different tools used on different platforms, and two to three weeks are dedicated to teaching tool usage. It is only after the student gains a certain level of skill with the tools that they can begin to build the skills required for job functions, and a significant amount of cyber warfighting knowledge is gained later through on-the-job training and real-world experience. Therefore, a single interface that is both tool- and platform-agnostic would reduce the amount of time it takes a student to become proficient at operating, which would reduce training costs and permit more time to be dedicated to scenario-based training.

USER ANALYSIS DRIVEN DESIGN

Following general GUI and user-centered design principles (Constantine and Lockwood, 1999; Norman, 2013; Tufte, 1990; Abras, Maloney-Krichmar, & Preece, 2004; and Sharp. Rogers, & Preece, 2007), to design a single next-generation interface to improve OCO, the most common behaviors that operators perform were analyzed to define workflows, and these workflows became the use cases for our design research and prototyping.

Workflows

To determine system requirements, common workflows were identified to provide use cases. For all workflows, the operator is the primary end user and performs all functions. Table 1 lists the use cases identified for the research.

Table 1. Operator Use Cases

Use Case 1: Identify connected target hosts and select one for action
1. The operator selects the endpoint to connect.
2. The operator creates a connection and the system connects.
3. The operator adds the endpoint to the working set.
Use Case 2: Identify hosts on the target network.
1. The operator selects the node to conduct the survey from.
2. The operator invokes the survey.
3. The system conducts the necessary commands (ARP, ping, NetBIOS, partial handshake).
4. New nodes are identified and displayed in the interface.
Use Case 3: Exploit a new host.
1. From the interface, the operator selects the target to exploit.

2. The interface highlights usable exploits in priority of use.
3. The operator clicks the exploit and the system exploits the target.
Use Case 4: Switch target context.
1. From the representation of the network, the operator knows on which target hosts he currently has an established session.
2. The operator selects a different host and all commands sent from the Heads Up Display (HUD) now reference the target.
3. The operator can select multiple targets and send the HUD commands to them all.
Use Case 5: Perform an initial status check.
1. Once a session is established to a target for the first time, conduct an initial status check (survey).
2. The survey returns the status of the target to the operator.
3. The defensive posture is automatically checked against the current RoE, and commands/functions that are not permitted are greyed out.
Use Case 6: Collect information from the target.
1. In an established session with the target, the operator collects information from the target.
2. Platform-specific options (e.g., Registry query for Windows) are displayed in the command options.
Use Case 7: Turn on/off layers of information.
1. The operator can turn on different layers of information
2. Example information: Identify targets that are hazardous, See where all tools were last installed, Identify targets of interest, Show previous actions (slug trail).
Use Case 8: Replay actions within the HUD.
1. The operator can reference a dynamic timeline that logs all actions performed during the session.
2. The operator can access/search the timeline for previous actions and data.
Use Case 9: Switch tool context.
1. The operator can select different tools in the HUD, which change the context of the menu options.
2. The operator can switch back and forth quickly

GAME-BASED HARDWARE AND SOFTWARE CONSIDERATIONS

Natural Interface Prototype: Principles Underlying the Design

We turned to game designers for GUI inspiration because unlike traditional software programs or apps which are often able to require users to learn a less intuitive GUI due to their utility, games must be walk-up-and-play intuitive to succeed because players are typically unlikely to continue playing a game with a less-than-perfect user interface (Stapleton, 2015). Video game design presents as the ideal benchmark to assess the impact of interface on user experience.

Based upon our understanding of the needs of the operators, a front-end game-inspired visual interface for OCO was designed centered on three primary goals: (1) reduce the complexity of the underlying tool, (2) improve situational awareness, and (3) reduce human error. We refer to this project as CyHUD and will use this name throughout the rest of the paper for ease of reference.

The key requirement was to give operators an interface to interact with underlying network agents and topologies through a range of different presentation and interaction models. These models needed several common core abilities, and must allow an operator to access them from different workbenches. The interface must also be usable across a range of target hardware configurations including various monitors, mouse and keyboard, touch-screen laptops or computers, monitor displays with Kinect-style gesture sensors (http://www.xbox.com/en-US/xbox-360/accessories/kinect), matrixed and dynamically deployable monitor arrays, large touch surfaces, and an Oculus Rift virtual reality headset (https://www.oculus.com/en-us/rift/) and Haptix/Touch+ 3D multi-touch station (http://www.ractiv.com/touch.html).

To design this new system, the team researched a series of interrelated components, including:

- Presentation Platforms: For the initial prototype, this is the computer and touch-screen monitor, but in the future it may include the Oculus or other displays (the hardware and software that deliver the visualizations).

- Workbench: The role-oriented presentation view composed of the presentation and interaction components that can be used for specific purposes to visualize or work with a dataset.

- Data Utilities: The tool or interface that links a workbench to the target data domain (or imports a database into a workbench).

- Interface Standard: The visual/intellectual construct that shows and makes the data intelligible. This is the view of the data presented by the workbench.

- Interface Paradigm: The conceptual way someone interacts with data through an interface standard; includes concepts like “select” and “move” and “delete”.

- User Interface (UI): The actual implementation of the Interface Paradigm (e.g., an operator “selects” by using the mouse to click on a point of data). The UI is often unique to the presentation layer and the workbench. And sometimes it is unique (and custom) to all portions of the system.

- UI Component Layout: The user-customized settings for a specific UI and/or Workbench.

- Optional Interface Linkage: To make the UIs more reusable, they can have a separate way of linking to the different presentation platforms and workbenches.

The interface design approach uses both 2D and 3D displays. Standard representations of datasets and events rely on a large set of 2D presentation and interaction elements and these are included in the system workbenches, but they are also augmented by 3D presentation of topologies. Traditional 2D representations of data are coupled with 3D game and simulation techniques to offer new n-dimensional representations of target data domain. This allows different elements of the environment to be mapped to different visual objects in the presentation. This provides an important perceptual change from the conventional concepts of the 2D standard GUI.

Offering both 2D and 3D views allows users to perceive and understand the underlying complex data sets in emergent, exploratory ways. These views allow users to apply segments of the data sets, models, attributes and related events to different axes or visual components in the display domain, creating new types of visualizations. Previous efforts to map complex network operations to multi-dimensional views (e.g., CAIDA Walrus http://www.caida.org/tools/visualization/walrus/, NICT Nicter http://www.nicter.jp/nw_public/scripts/atlas.php) have only hinted at the true flexibility and adaptability featured in CyHUD’s visual presentations.

CyHUD Design

The system does not offer a single defined screen layout, but instead it allows operators to customize its highly flexible components as needed. A flexible layout also allows for a more flexible work environment, using different hardware and software elements however they can best contribute to mission success. For example, a video or touch wall allows for greater collaborative work than an individual immersive setup such as the Oculus Rift.

In addition to a fully customizable and mappable presentation component for viewing data, CyHUD offers utilities to easily map datasets in the common usage domain. Templates for common visualizations are provided and sharable across user workbench instances, and the UI will be command set agnostic. The design contains a set of user friendly workflow macros and the ability for the user to predesignate their own macros. The macros represent commonly executed actions that are subject to human error or that are repetitive, such as perform security checks on a system, or collect specific data from a hard drive. User defined macros allow users to customize macros for specific command chains speeding up the process and reducing the chance of human error in repetitive commands.

An important element of UI component and template design and development is validating whether a particular component or template can appropriately reflect properties and models within the system environment. In many design efforts, there is a tendency to over-design elements to "look cooler" or just to use 3D elements. This was studiously avoided, and each CyHUD element reflects critical data in the model/property/event, and presents this data to the user in an intuitive and efficient manner. In any workbench, the operator can select those components that best serve their perceptions, abilities, or needs.

Super-users and SMEs contributed to the design process from the earliest stages. The UI design approach included extensive interviews with nine client-identified key users and OCO experts to perform task analysis, which was used to determine what separates experts from less effective users. This analysis was used as the basis for creating UI elements and the different contexts that drive UI presentation. The design process extended beyond just providing broad customization and instead focused on where to offer easy customization. In future system versions, we could also provide extensibility to support user customization and pursue the complex UIs seen in expert EVE Online and World of Warcraft player screenshots, often considered the gold standard for game-based UIs for complex systems.

This design approach resulted in an interface that reflects the critical data present in the model/property/event, and presents this data to the user in an intuitive and efficient manner. Several CyHUD design elements are discussed below to illustrate the interface.

In CyHUD, each target system that the user interacts with is presented as a 3D system box (see Figure 5). This box displays information about the system as it is known, and all aspects of the box convey data; for example, the box edge color indicates the system state (whether it is currently selected, normal, or only has historical information available). The top face of the cube indicates the target's Operating System (OS) (if known); the left face of the cube shows the number of networks that the system is a part of; and the right face of the cube is color coded to indicate whether this system is part of the network currently shown in detail on the Network view. For the prototype no disabilities are accommodated.

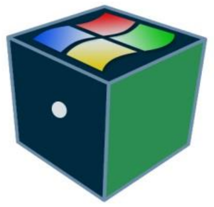


Figure 5. 3D system box representation of a host

Around the system box is an Information Fan of eight segments (see Figure 6). Each segment contains critical information about the system that is important for maintaining situational awareness and danger levels. There are eight symbols used to show vital information about the system (malware, anti-virus, Typhoon, user activity, target, monitoring, key terrain, and alert). The color of the background/icon represents the last known state of the symbol. Symbols are always displayed in the same relative location on the fan, as shown in Figure 6. An empty fan slice with no symbol or color indicates that no information is available about that aspect of the machine. (It does not indicate that danger is not present, only that there is currently no information). This Information Fan only appears around systems that have active sessions running on them. The Information Fan of a non-selected box is desaturated. If any alerts appear on one of these boxes, the correct segment of the fan will change to full color.

Selecting the box opens up a Command Fan (see Figure 7) that displays all available commands for that box in its current state. Previously, commands all had to be memorized and recalled paving the way for user error. The Command Fan provides contextually relevant command options and is automatically pre-populated with any known system information.

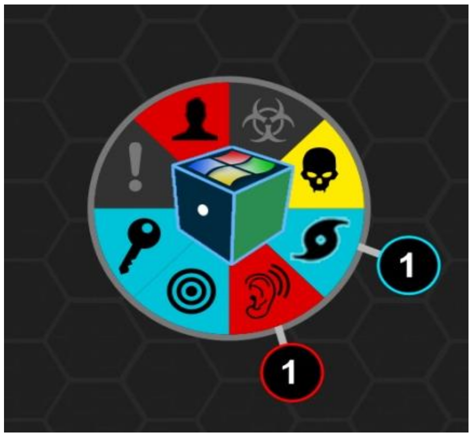


Figure 6. Information Fan

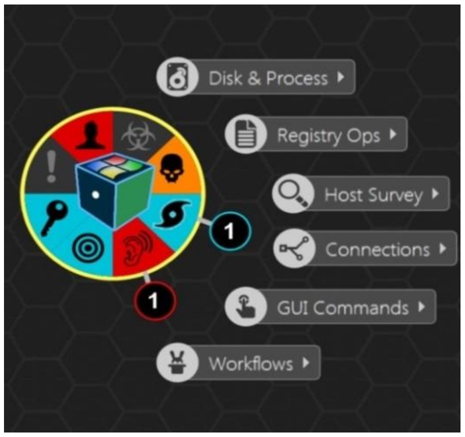
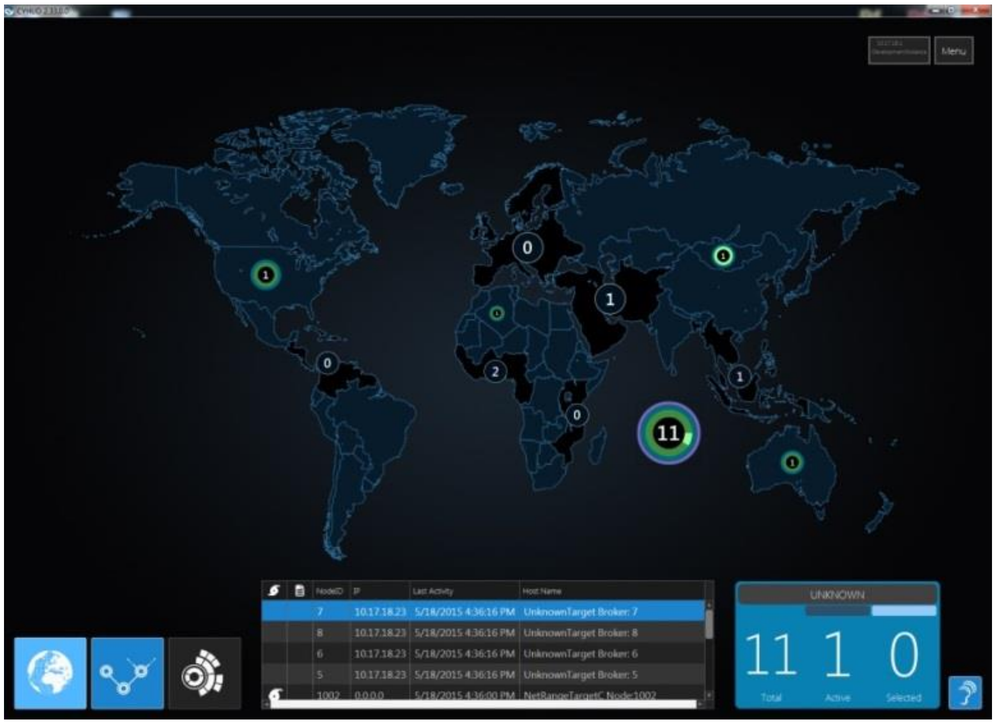


Figure 7. Command Fan

The system allows multiple views that each support unique operations within the current situation. The World View (see Figure 8) is used to route the session through intermediary computers around the world to hide the actual origin of the attack. The World View collapses data into regions to provide a high level picture of the nodes active in routing, and indicates whether the source machine issuing commands or redirector are located within the region. No interaction with individual systems occurs in the World View and no connections are illustrated here. The World View provides situational awareness throughout the attack by indicating status changes on utilized machines.



		NodeID	IP	Last Activity	Host Name
		7	10.17.18.23	5/18/2015 4:36:16 PM	UnknownTarget Broker: 7
		8	10.17.18.23	5/18/2015 4:36:16 PM	UnknownTarget Broker: 8
		6	10.17.18.23	5/18/2015 4:36:16 PM	UnknownTarget Broker: 6
		5	10.17.18.23	5/18/2015 4:36:16 PM	UnknownTarget Broker: 5
		1002	0.0.0.0	5/18/2015 4:36:00 PM	NetRangeTargetC Node:1002

Figure 8. World View

The Network View (see Figure 9) provides a numeric presentation of information known to the user about the network. This information is presented as a series of concentric rings of information, allowing the user to narrow down choices until viable targets are identified.

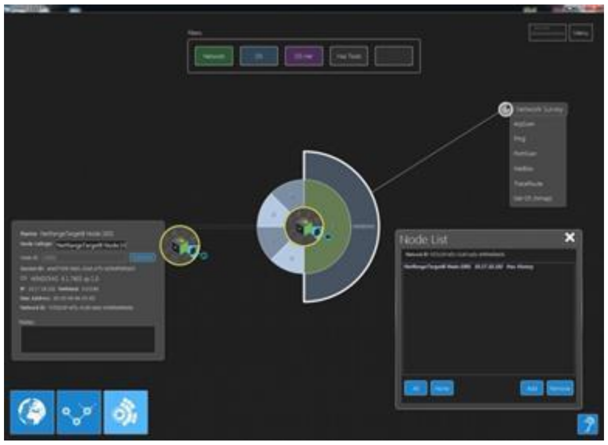


Figure 9. Network View

The Topology View (see Figure 10) is the workspace where users interact with the individual systems, it presents a node centric view of the operator’s actions. It displays information about each node and the tunnels the user used to reach them. It shows the nodes of the workset in a manner to enhance the user’s situational awareness of every system they have touched. There may be hundreds of nodes in a network, only those with active sessions, historical data, or selected from the Network View are displayed.

Operators may select any node with a live session and make that node an Active Session. This opens the Information Window for the node and uses a color code to ring the node. A History Window can be opened to view information on all actions taken by the operator. The system allows multiple levels or views that each support unique operations within the current situation.

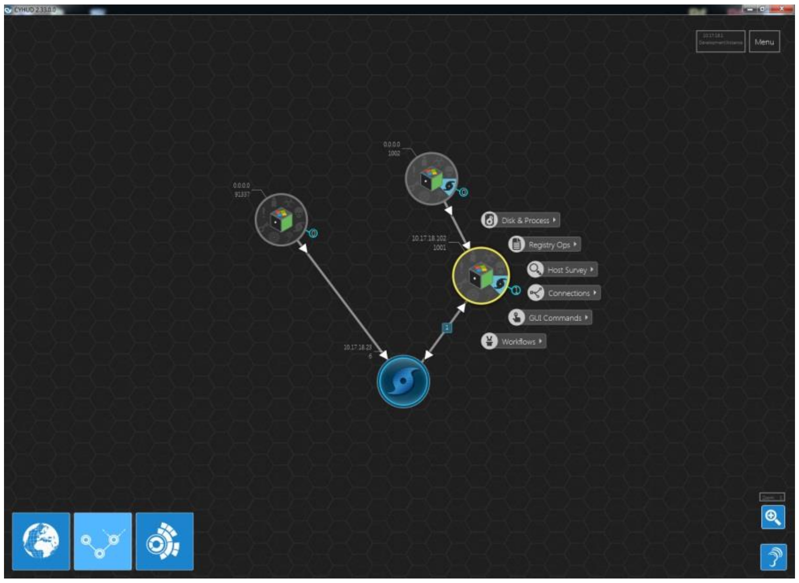


Figure 10. Topology View

Interaction Commands

Initially there was interest in gesture control and Oculus Rift as components of the solution. Informal user testing indicated that the gesture control was not yet mature enough for integration and that wearing the current Oculus Rift for the long session time an operator would use the system was not practical. However, in the current system, operators can interact with CyHUD using common touch commands and simple voice commands:

- Touch – Press detection. The finger may be lifted or held in place.
- Tap – Press one finger and release quickly.
- Hold – Press one finger and keep finger in place for 0.75 seconds.
- Hold and Tap – Hold one finger in place and then tap with another finger.
- Drag – Press one finger on a location and then move the finger elsewhere before releasing.
- Flick or Swipe – Press one finger on a location and then move the finger away quickly.
- Pinch – Press two fingers to the screen and move them closer together.
- Spread – Press two fingers to the screen and move them farther apart.

INFORMAL PLAYTEST

CyHUD has demonstrated its capabilities at the national-level cyber exercise Cyber Flag, and it is currently undergoing continuous playtesting with users at the KEYW training facility. To capture the results of this research on the prototype interface, the playtest experiences focus on workflows based on use cases 1–7 provided in Table 1 Operator Use Cases. For the first playtest, fifteen Subject Matter Expert (SME) users were selected, each with over two years of experience in cyber operations and cyber capability development. These users received approximately 10 minutes of instruction on the CyHUD interface and then worked through the common use cases for an interactive operator, and described their experiences by completing a twenty-question survey. Their CyHUD stations included a 24-inch touch-screen monitor, mouse, and keyboard.

The playtesters explored the three main goals of CyHUD: (1) to increase situational awareness, (2) to reduce human error, and (3) to reduce training time. The majority of the testers (87%) felt that CyHUD did increase their situational awareness, a finding that was particularly noted in the session topology view where the users could send commands to nodes and observe data routing between those nodes. In addition, a majority of the testers (77%) thought that CyHUD would reduce human error, especially in comparison with other tools that they had previously used. They attributed this finding to the interface's use of data manipulation and form input using a one-click/touch approach, which minimizes the need to type IP addresses. Lastly, a majority of the testers (88%) believed that CyHUD would reduce training time; in particular, they noted that the user-friendly workflows and clear information story presented enabled them to quickly acclimate to the tool (within 50 minutes on average).

The user experience surveys revealed some opportunities to improve the current design:

1. Remove the need to set source flags: testers did not like setting source flags on nodes before executing commands; they assumed that when they touched or clicked on a node that the system would automatically change its context.
2. Define symbols: testers noted that tool tips and legends to define symbols would be helpful.
3. Improve touch and voice input interface capabilities: testers did use the touch interface to manipulate nodes and to zoom in and out, but they switched to the keyboard and mouse to address textual information as the scroll bars were difficult to navigate by touch. Many testers wanted to manipulate the data beyond the capabilities currently offered, and while they could copy and paste content, they could not transfer their work to data manipulation fields solely with the touch-screen. While the prototype made use of a multi-touch interface and voice activated input. User testing discovered areas where the design can gain efficiency in maximizing the multi-touch experience and augment complex input with speech-to-text.
4. Provide a more holistic view: the playtesters also stated that they wanted to see all of the target attributes they had discovered from the commands they had run. Such a holistic view is not available in many tools, and CyHUD did not offer this view at the time of the playtest.
5. Accommodate experienced users: experienced users identified use cases which were not accounted for under the initial design. For example, an experienced user may wish to use the tunnel session established with the underlying tools to route their browser connections into the internal target network. The current design does not account for this.

CONCLUSION AND FUTURE WORK

The informal playtest produced valuable user feedback indicating how effectively CyHUD allows users to transition from the operational tasks of commanding tools and navigating targets to the analytical tasks of data synthesis and decision-making. Thus, as we continue to optimize the workflows and situational awareness, we will also pursue more advanced interfaces for data analysis in future releases of CyHUD. It is clear that moving beyond the standard CLI and GUI systems currently available was intriguing to the end users who experienced the system.

As designed, the CyHUD prototype satisfies much of the basic functionality required to perform a cyber-operation, however in order to operationalize the current prototype the following actions would be necessary:

1. Reexamine the exploitation use case: the means in which controls over exploitation were implement need to be more robust. The interface should work with the underlining tools and handle repetitive configuration tasks for the user.
2. Address operational use cases that surfaced during testing that were not initially accommodated.
3. Research and implement potential optimization gains with better use of multi-touch and voice interaction.

As this remains a new concept in the OCO community, end users and decision makers are initially resistant to the idea. But having the ability to share a prototype system with positive initial playtest results will ideally pave the way for continued research and development of the CyHUD tool driven by the end user community.

ACKNOWLEDGEMENTS

We would like to acknowledge the work of the KEYW staff members, OCO operators, and instructors who shaped the design of CyHUD and served as playtesters and the design and development team at BreakAway Games for making the vision a reality.

REFERENCES

Abras, C., Maloney-Krichmar, D., Preece, J. (2004) User-Centered Design. in Bainbridge, W. (Ed.) *Encyclopedia of Human-Computer Interaction*. Thousand Oaks, CA: Sage Publications.

CAIDA Walrus http://www.caida.org/tools/visualization/walrus/

Constantine, L., and Lockwood, L. (1999). *Software for Use: A Practical Guide to the Essential Models and Methods of Usage-Centered Design*. Reading, MA: Addison-Wesley.

Hutchins, E. M., Cloppert, M. J., and Amin, R. H. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Paper presented at the 6th Annual International Conference on Information Warfare and Security, Washington, DC, 2011. Retrievable here: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf

Kinect information can be accessed here: http://www.xbox.com/en-US/xbox-360/accessories/kinect

Metasploit Unleashed: Mastering the Framework, Offensive Security.com, last modified October 15, 2010, accessed June 06, 2011, http://www.offensive-security.com/metasploitunleashed/Metasploit_Unleashed_Information_Security_Training.

NICT Nicter http://www.nicter.jp/nw_public/scripts/atlas.php

Nmap Network Scanning, the nmap.org web site, accessed May 31, 2011, http://nmap.org/book

Norman, D. A. (2013). *The Design of Everyday Things: Revised and Expanded Edition.* New York: NY: Basic Books.

Oculus Rift information can be accessed here: https://www.oculus.com/en-us/rift/

Sharp. H., Rogers, Y. & Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction*, 2nd ed. Hoboken: NJ: John Wiley & Sons Ltd.

Stapleton, L. (2015). How UX Design Affects the Success of Video Games. Retrieved 8/20/2015 from: http://www.dtelepathy.com/blog/design/the-importance-of-ux-design-expanding-audience-of-video-games

Touch+ information can be accessed here: http://www.ractiv.com/touch.html

Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT:,Graphics Press.

U.S. Joint Chiefs of Staff. Cyberspace Operations. Joint Publication 3-12(R). Washington, DC: U.S. Joint Chiefs of Staff, February 5, 2013. Accessible here: http://www.dtic.mil/doctrine/new_pubs/jp3_12R.pdf