

# The Crossed Swords wargame: Catching NATO red teams with cyber deception



© 2015-2018 Cymmetria® Inc. All rights reserved.

## BACKSTORY

Once a year, the pentesters\* and red teams of the countries of NATO descend on Tallinn to visit the NATO Cooperative Cyber Defence Centre of Excellence (CCD COE) for the Crossed Swords wargame (not to be confused with [Locked Shields](#)). They are given an objective and initial access to the wargame environment, and are then set loose on the network. The goal is to help the red team improve by having a yellow team monitor the network and show the pentesters how they are caught along the way.

Once in a while, a vendor is invited to join the yellow team and is strictly warned: no vendor has been successful at catching the red team yet. Security products are wired to catch attack tools. “When it comes to humans,” explained Hillar Aareleid “products fail miserably.”

Cymmetria had one job: ambush the pentesters and catch them red-handed.

## GROUND RULES

The rules we were given for our participation in Crossed Swords were simple: we could use MazeRunner to create any asset we wanted—decoy machines, “breadcrumb” files, etc.—in order to detect where and how the pentesters attacked. Besides MazeRunner, normal honeypots from several popular open source projects were also deployed (in equal numbers).

Since the goal of the exercise was to help the attackers improve, we provided them with the knowledge that MazeRunner was in the network, and also showed them in real time whenever they stepped into a trap.

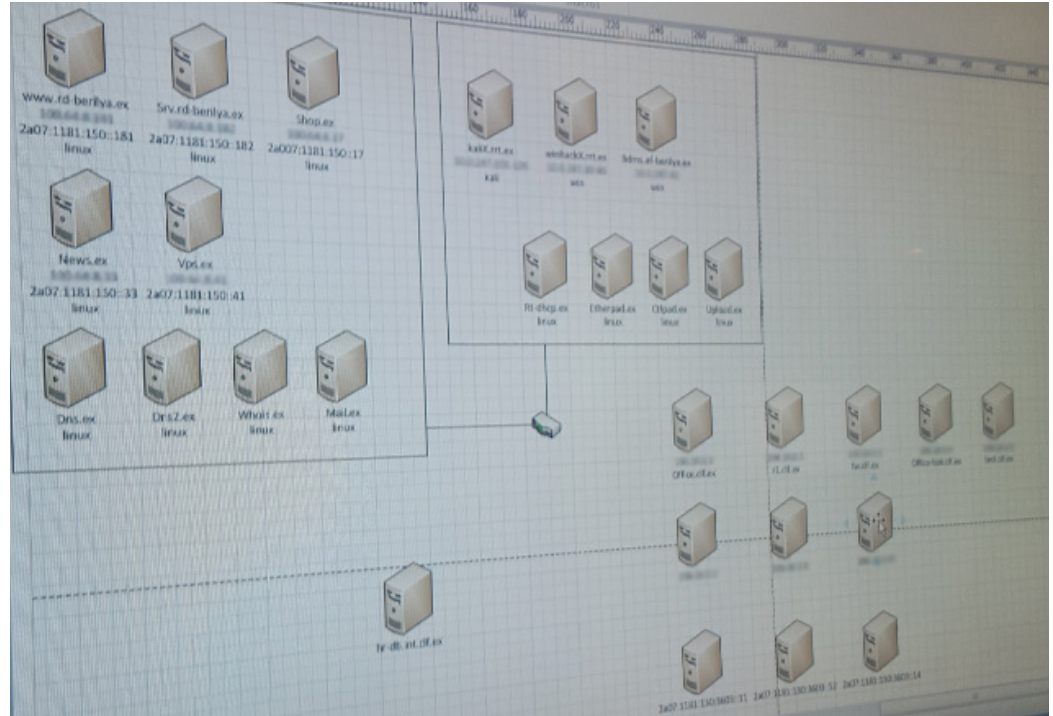
The attackers marked machines on which they had root or administrator capabilities as red, and those on which they had regular access as yellow.

Besides hacking in the technical sense, the red team also had other capabilities—including a Special Forces team that played the wargame by breaking into a hotel room and copying a laptop’s hard drive contents.



Special Forces copying a laptop hard drive

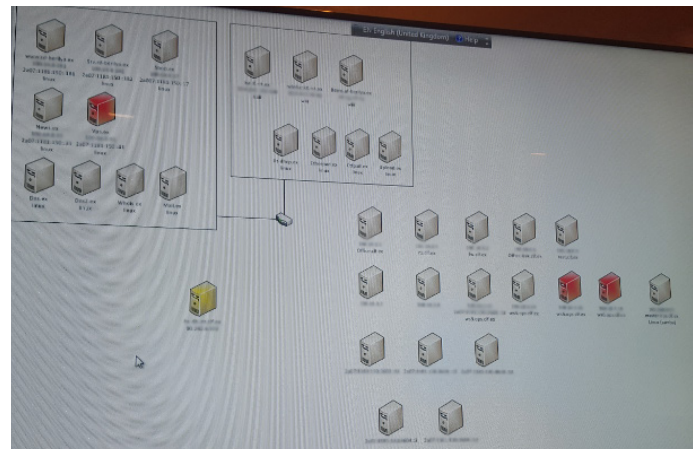
Like any military operation, the attackers began with intelligence collection. They were given information on one half of the network, and did a quiet port scan to find a few more machines (among those were some of our



Initial attackers' mapping of the network

decoys). The initial breach was through a vulnerable server that they compromised using a public exploit. Afterwards, they compromised two endpoint machines through spear phishing. They looked for credentials on the machines, which is when they picked up MazeRunner breadcrumbs (MazeRunner breadcrumbs are pieces of data to be used in lateral movement, such as credentials, cookies, configuration files, etc.). The third machine they accessed was one of our decoys, which was set up to look like an HR database.

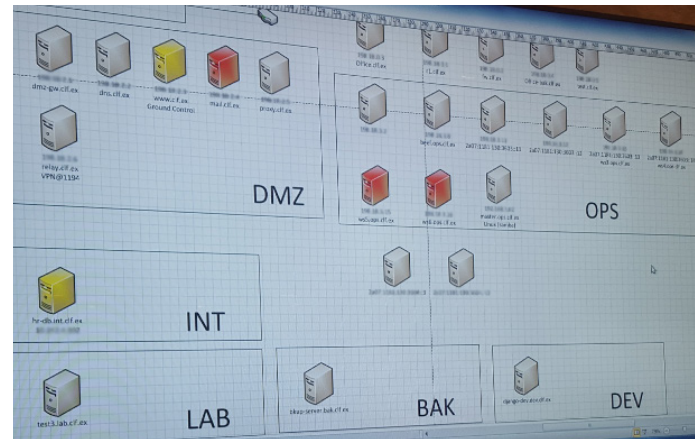
In the following image, you can see our decoy in yellow (meaning the pentesters were able to "compromise" it, thinking it was a regular server on the network). On the compromised decoy, the attackers found user credentials leading to a MySQL database, which was running on yet another decoy.



First compromised machines (decoy in yellow)

We were there to help, but this was still a wargame. Few things are as satisfying as seeing attackers write “We got the creds for the relay machine on workstation 6, infect,” when it’s exactly where you want them to go.

Since our breadcrumbs were the first intelligence the attackers secured in the post-exploitation phase, their path was straightforward. They mapped the new networks they discovered based on our deceptive data, discovering decoys as they worked.



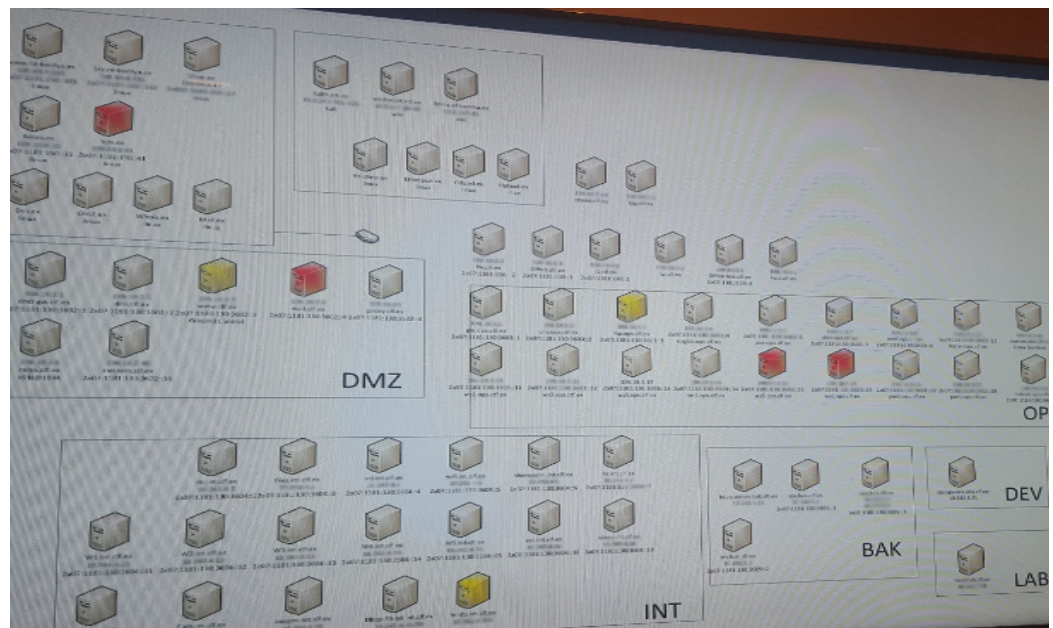
New networks mapped out, with all discovered machines being decoys

Then, to our surprise, our decoys stopped communicating with the management server.

The red team’s network team leader (who was in-the-know on which deception elements were in play) came over to us and said “My guys are running on one of your decoys but I don’t see it on the alert screen. Is everything working?”

Turns out that most of our decoys were having a networking issue connecting back to the management server. This was bad, and we soon uncovered the reason: the attackers were ARP poisoning the environment so aggressively that SSL traffic was not possible in certain connections. Once we realized what was going on, we locked down the ARP tables and made sure the decoys could continue communicating back to the management server.

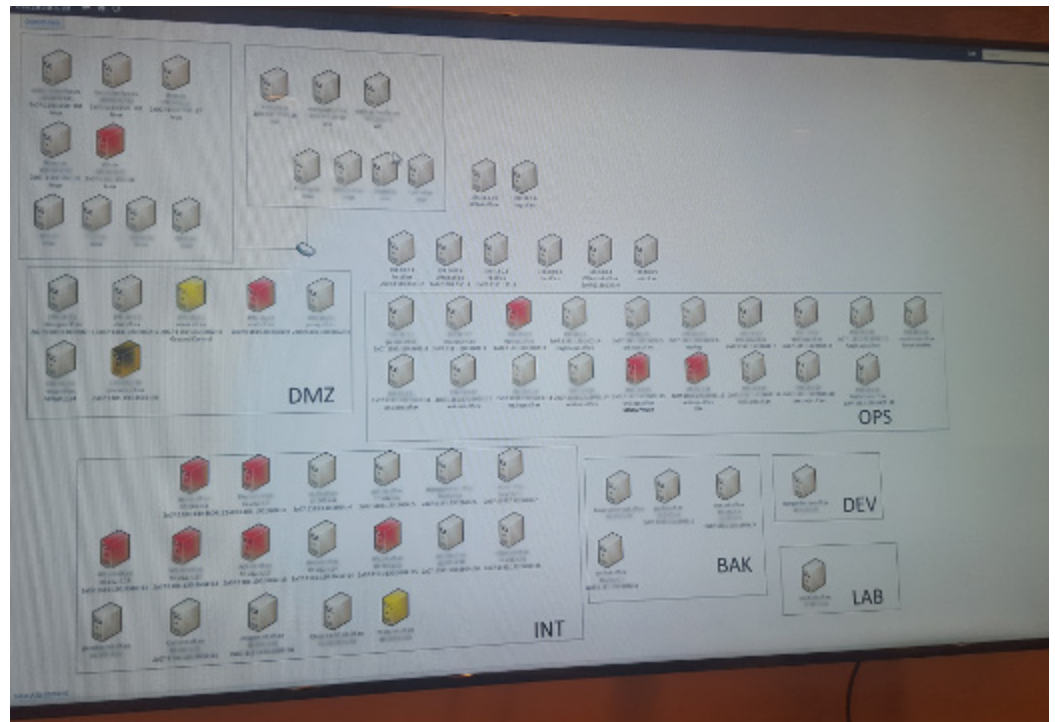
Later on in the day, the attackers were able to make DNS zone transfer work, and now had a map for the rest of the network.



DNS zone transfer used in mapping phase

With the first day of the exercise over and the network mostly mapped out, the attackers' intention was now to start making major progress on compromising the network, advancing on their objective. Their client-side team was working hard to get the credentials to a domain controller in the INT (intelligence) network, running mimikatz and other post-exploitation tools, looking into any content they could find to locate credentials, and trying golden ticket attacks against the domain controller.

Once the domain controller was compromised, the whole network fell like a house of cards.



Once the domain controller was compromised, the whole network fell like a house of cards

Notice the black and yellow machine in the network map. Just like in nature, those colors signify danger, and that's the color scheme the attackers used to identify honeypots they spotted (all of which were open-source honeypots).

Showing the attackers how they were caught during the first day made them more aware of the traps. Despite having this knowledge, the red team still didn't pinpoint any of the decoys, including the machine they were running on during the first day.

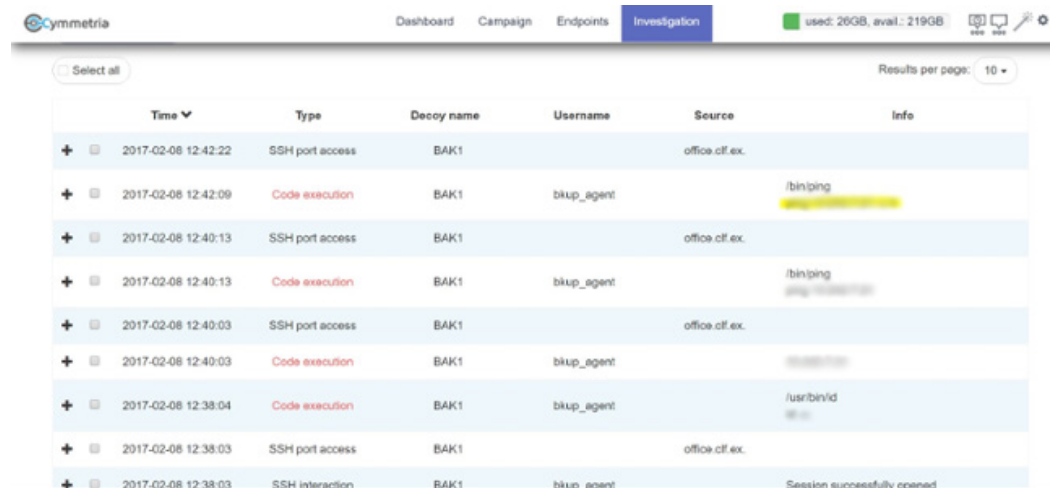
After owning INT, the attackers started looking into the other networks. BAK (the backup network) was the first one they went after. Since the decoys were the first machines they discovered during the recon phase on that network, they attacked those first.

	Time	Type	Decoy name	Username	Source	Info
+	2017-02-08 09:44:20	Code execution	BAK1	bkup_agent		/usr/bin/id -u
+	2017-02-08 09:44:18	SSH interaction	BAK1	bkup_agent		Session successfully opened
+	2017-02-08 09:44:18	Code execution	BAK1	bkup_agent		/bin/hostname hostname
+	2017-02-08 09:44:18	SSH port access	BAK1			
+	2017-02-08 09:40:56	SSH port access	BAK1			
+	2017-02-08 09:40:51	SSH port access	BAK1			
+	2017-02-08 09:40:51	SSH interaction	BAK1	bkup-agent		Invalid user tried to connect (before enteri...

### Screenshot of an attacker connecting through SSH to a decoy on BAK

An operational mistake the attackers routinely made was to use an exploited endpoint to see if they were able to connect to a machine, but then use their own machine afterwards. This provided us not only with the intelligence on the source address of the exploited hosts, but also with the addresses of the attackers' machines.

The attackers then used the machine again in an attempt to see if it was connected to the DEV (development) network, and tried to ping the host that they discovered there (you guessed it, another decoy).



	Time	Type	Decoy name	Username	Source	Info
+	2017-02-08 12:42:22	SSH port access	BAK1		office.cif.ex.	
+	2017-02-08 12:42:09	Code execution	BAK1	bkup_agent		/bin/ping
+	2017-02-08 12:40:13	SSH port access	BAK1		office.cif.ex.	
+	2017-02-08 12:40:13	Code execution	BAK1	bkup_agent		/bin/ping
+	2017-02-08 12:40:03	SSH port access	BAK1		office.cif.ex.	
+	2017-02-08 12:40:03	Code execution	BAK1	bkup_agent		
+	2017-02-08 12:38:04	Code execution	BAK1	bkup_agent		/usr/bin/id
+	2017-02-08 12:38:03	SSH port access	BAK1		office.cif.ex.	
+	2017-02-08 12:38:03	SSH interaction	BAK1	bkup_agent		Session successfully opened

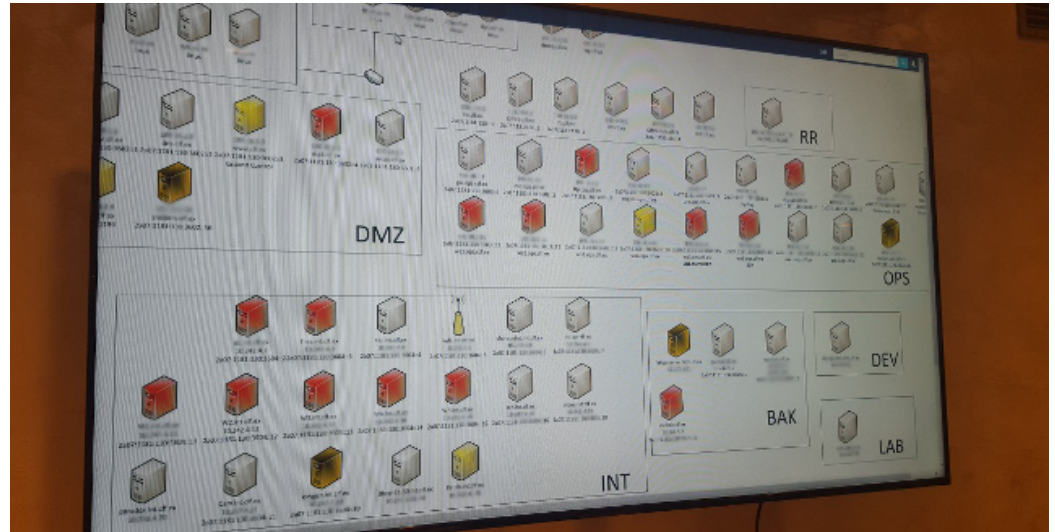
### Attackers checking if the compromised machine has access to the DEV network

APPROACHING  
THE FINISH  
LINE

It was the third and final day of the exercise, and the attackers were a significant distance from their target. They decided to step it up a notch and risk further detection.

We saw several interesting things by the end of the time limit. After detecting three more regular, open-source honeypots in the network, the attackers used one of our decoys as their staging ground for lateral movement—and we decided to allow it rather than automatically kicking them out.

They did everything they could to get to the DEV and LAB networks. One of the attackers started researching the backup decoy and was suspicious of what its role was in the network. Then, on a hunch, he started correlating between the MazeRunner Investigation screen (which was visible throughout the exercise, minus the IP addresses), and the commands he was running on the decoy. This was the first and only MazeRunner decoy that was detected throughout the exercise.



Honeypots and a MazeRunner decoy, marked in yellow and black

The web hacking team spent a lot of time trying to crack the first decoy that they had encountered: the HR database. We saw them try to authenticate using all the credentials they had collected, and then run a lot of different queries and GET requests to try to hack the decoy. Throughout the exercise, they did not give up. This would prove very valuable for defenders in a real-world scenario, as we not only gained clear intel from the attackers' actions, it also wasted their time and resources throughout the week.

Time	Type	Decoy name	Username	Source	Info
2017-02-09 10:13:46	MySQL request	INT1	test	10.10.10.10	Access denied on connect
2017-02-09 10:13:44	MySQL request	INT1	scoringbot	10.10.10.10	Access denied on connect
2017-02-09 10:13:42	MySQL request	INT1	sarah.porter	10.10.10.10	Access denied on connect
2017-02-09 10:13:40	MySQL request	INT1	oracle	10.10.10.10	Access denied on connect
2017-02-09 10:13:39	MySQL request	INT1	operator	10.10.10.10	Access denied on connect
2017-02-09 10:13:36	MySQL request	INT1	morris.beck	10.10.10.10	Access denied on connect
2017-02-09 10:13:36	MySQL request	INT1	morris.beck	10.10.10.10	Access denied on connect
2017-02-09 10:13:34	MySQL request	INT1	mona.gray	10.10.10.10	Access denied on connect
2017-02-09 10:13:32	MySQL request	INT1	madonna.madonna	10.10.10.10	Access denied on connect
2017-02-09 10:13:29	MySQL request	INT1	mad.prof	10.10.10.10	Access denied on connect

We were able to capture all the compromised credentials

+ [ ]	2017-02-09 10:53:06	SSH port access	INT1		TEMP-PC	
+ [ ]	2017-02-09 10:48:17	MySQL request	INT1	contracts	TEMP-PC	FLUSH PRIVILEGES
+ [ ]	2017-02-09 10:44:23	MySQL request	INT1	root	TEMP-PC	Access denied on connect
+ [ ]	2017-02-09 10:44:15	MySQL request	INT1	contracts	TEMP-PC	Quit
+ [ ]	2017-02-09 10:43:59	MySQL request	INT1	contracts	TEMP-PC	UPDATE user SET 'Host' = '%' WHERE Us...
+ [ ]	2017-02-09 10:41:29	MySQL request	INT1	contracts	TEMP-PC	SELECT * FROM user
+ [ ]	2017-02-09 10:41:24	MySQL request	INT1	contracts	TEMP-PC	SELECT * FROM users
+ [ ]	2017-02-09 10:41:15	MySQL request	INT1	contracts	TEMP-PC	use mysql
+ [ ]	2017-02-09 10:41:11	MySQL request	INT1	contracts	TEMP-PC	use Access denied for user 'contracts'@'%...
+ [ ]	2017-02-09 10:41:06	MySQL request	INT1	contracts	TEMP-PC	Session started

### Executing different queries

Before the end of the exercise, just as they were about to hit their objective, the attackers ran one last scan in one of the networks, and found something interesting:

```
(2:10:59 PM [redacted] entered the room.
(2:12:22 PM [redacted] found a new host on the DEV zone: 6.22. According to the certificate on port 443, the issuer is Cymmetria. So it seems this is a Honeypot. Or at least the service. Please add it the map.
```

### Attacker chat screenshot

“Found a new host on the DEV zone: 6.22. According to the certificate on port 443, the issuer is Cymmetria. So it seems this is a Honeypot. Or at least the service. Please add it [to] the map.”

This specifically wasn't a decoy—it was our management server. We have since learned from this and added the ability to put in a custom certificate or change the issuer, in order to make it harder to identify our management server using this technique. It's a shame really; they could have taken another step and understood which asset they had discovered, but they gave up when they saw the certificate.

## POST-GAME RECAP

To summarize, it was extremely interesting to see how a group of digital battlefield experts operated, worked, and most of all how they handled cyber deception in a real-life scenario with an objective and a deadline. We are very thankful for having the opportunity to work with such a skilled and talented group of individuals who were amazing in their roles as both attackers and defenders.

For our part, we learned many things that will improve our ability to detect and stop the best penetration testers and red teams in the world.

**For more information about MazeRunner, or for a product demonstration, please contact Cymmetria at [info@cymmetria.com](mailto:info@cymmetria.com).**