Securing Web Resources using RFID, Dynamic Groups, SIEM Alerting and Honeypots

Mark Deegan
Dept. of Computing, Carlow Campus
South East Technological University
Rep. of Ireland
mark@deeganit.net

Martin Harrigan
Dept. of Computing, Carlow Campus
South East Technological University
Rep. of Ireland
martin.harrigan@setu.ie

***Abstract*—**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

II. Related Work

We categorise related work into four areas:

using RFID to combine physical with digital access [1] [2] [3] [4] [5] [6] [7]

dynamic role-based access control [8] [9] [10] (insider threats) [11]

Zero Trust Architecture (ZTA) [12] [13] [14] [15] [16] [17] [18]

III. Method

Our system comprises a variety of off-the-shelf components:

- Microsoft Active Directory with LDAP support
- Raspberry Pi connected to an RFID door entry system (see Fig. 2)
- Progress Kemp LoadMaster [19]
- Progress WhatsUp Gold [20]
- Web servers (Apache HTTP Server)

A. Microsoft Active Directory With LDAP Support

We used Microsoft Windows Server 2019 to run a Domain Controller (DC) (`dl.testlab.local`) for our domain (`testlab.local`). This is the primary DC for the scenario and it was setup to host Active Directory (AD) with LDAP support and the Domain Name System (DNS). To ensure that the requests for the Web resources went to the appropriate services on the Progress Kemp LoadMaster, we created corresponding DNS delegations. This is important as we wanted the external requests to be pointed to the external resources and the internal requests to be pointed to the internal resources. It also allows the LoadMaster to perform service health checks before responding to DNS requests thus preventing IP addresses being returned when resources are unavailable. We added two users, Jane Doe and John Doe, and two groups, Internal and External, to AD (see Fig. 1).

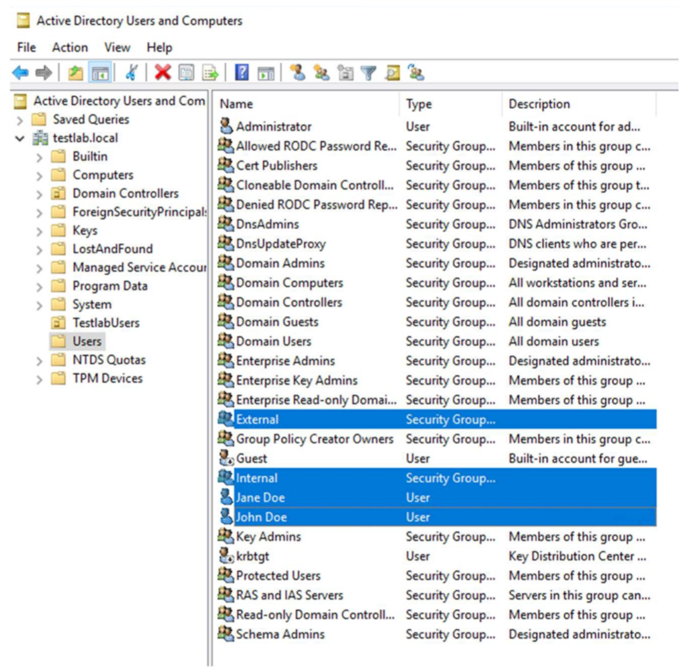

Fig. 1. We created two users, Jane Doe and John Doe, and two groups, Internal and External, in Active Directory.

B. Raspberry Pi and RFID Door Entry System

The Raspberry Pi is connected to an RFID door entry system. When a user performs a card swipe, a Python script running on the Raspberry Pi reads the identification details from the card and remotely changes the group membership of the user in AD. If the user is entering the building their group membership is changed from External to Internal; if they are leaving the building it is changed from Internal to External. It logs the event to a SIEM service (see Sect. III-E).

C. Progress Kemp LoadMaster

Progress Kemp LoadMaster (LM) is a reverse proxy and load balancer. It has many capabilities but for this scenario we are interested in the Edge Security Pack (ESP), the source IP blacklist from the GEO component, and the Web Application Firewall (WAF). We use the ESP for Single Sign-On (SSO) for

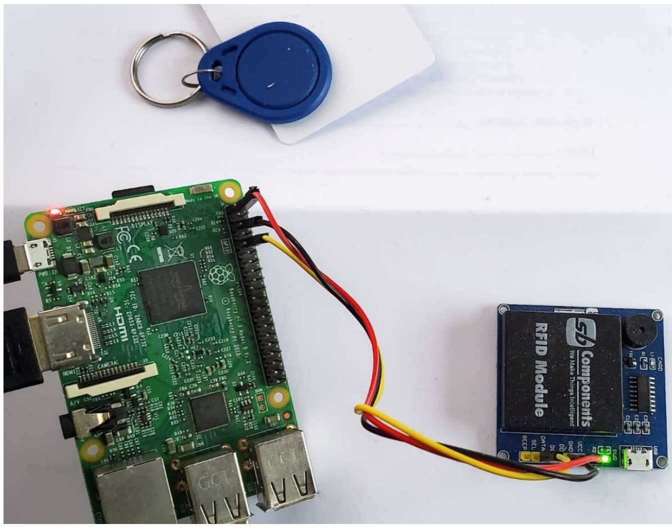

Fig. 2. The Raspberry Pi was connected to an RFID door entry system. We created a Python script that read the identification details from swiped cards and interacted with Active Directory.

HTTP(S) services and to communicate with the AD for both for logon and group memberships. This pre-authenticates a user before they gain access to a resource. We also enabled *group steering* on the ESP: this allows LM to send traffic to particular services based on their group membership in AD and goes beyond the normal use of groups to simply allow or deny access.

We created two steering groups associated with the Internal and External groups in AD. We created Perl Compatible Regular Expression (PCRE) rules to match the authorisation cookies and steer the requests to the appropriate services (see Fig. 3).

The login page that is displayed by the ESP is the same for both *valid* and *invalid* access attempts. A valid access attempt occurs when a user's group and request are both internal or both external; otherwise the access attempt is invalid. In both cases a log is sent to a SIEM service (see Sect. III-E). This helps with threat hunting as a threat actor will get the same login page for the honey pot as with the valid site. The honey pot can gather the details of the access attempt without being discovered.

The GEO component performs DNS resolution and service health checks before returning a result. We created GEO DNS entries for `internal.testlab.local` and `external.testlab.local` (see Fig. 4). We also used an IP blacklist that is updated daily to withhold DNS results from anyone on the list.

Additionally, we enabled the Intrusion Detection System (IDS) and Intrusion Prevention Systems (IPS) on LM. This includes rules defined by the SNORT community [21] and enables the SNORT rule filtering on the Layer 7 HTTP engine to check for any known bad requests. Figure 5 shows the configuration highlighted in green. There is also an option to prevent access via whitelists and blacklists highlighted in blue.

Finally, we enabled the Web Application Firewall (WAF) and the Open Web Application Security Project (OWASP) core rule set. This rule set performs anomaly scoring and identifies, for each request, the probability that it is malicious. The core rule set protects against SQL injection, cross-site scripting, remote code execution, buffer overflows, known vulnerabilities, and many other vectors of attack. We configured the WAF with source IP reputation blocking enabled which uses a global IP reputation list that is updated daily. Using the MaxMind [22] and the GEO component, it identifies the country of the source request and it can be configured to block specific countries or regions.

D. Web Servers

Our Web servers are hosted on virtual running Debian and a default installation of the Apache HTTP Server. The landing page is our "valid access" page and represents our secured Web resource. The "invalid access" page is served by a Flask application. It records the username and source IP of all requests and sends those details to a SIEM service (see next section).

E. Progress WhatsUp Gold

The Raspberry Pi, LM, and Flask application send logs to a SIEM service. We use Progress WhatsUp Gold (WUG) (see Fig. 6). In normal operation the events from the Raspberry Pi, RFID door entry system, and LM are logged. In cases where a user's credentials may be compromised, the Flask application logs an event with high priority.

IV. Results

We configured the system as described in the previous section. The goal of the study was to show the feasibility of the integration between the various components, and to demonstrate that dynamic group membership based on real-world location can secure Web resources. We performed two tests (Sect. IV-A and Sect. IV-B) to demonstrate this aspect of the system.

A. Internal User with External Threat

In the first case the user, John Doe, enters his office during normal working hours and swipes his access card at the door using an RFID tag. His group membership is set to Internal and the user can then access the resource internally. Meanwhile an external threat actor attempts to login from outside the office while the user is at work. They are denied access and they have their IP address logged to the SIEM service (WUG) as a breach attempt. This requires no extra overhead on the user to secure his credentials. The timeline of events is as follows:

1) John Doe enters his office and swipes his access card.
2) The user's group membership is changed from External to Internal. This event is logged to the SIEM service from the Raspberry Pi.
3) When the user gets to his desk they access the Web resource internally.

Fig. 3. ...

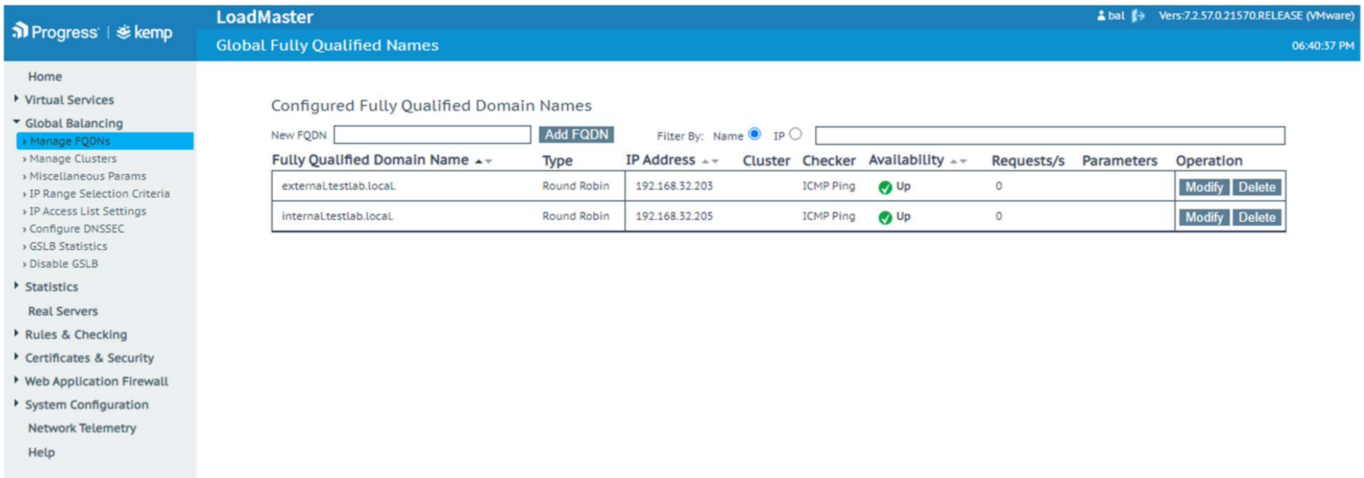

Fig. 4. We configured two FQDNs for `internal.testlab.local` and `external.testlab.local` using LoadMaster's GEO component.

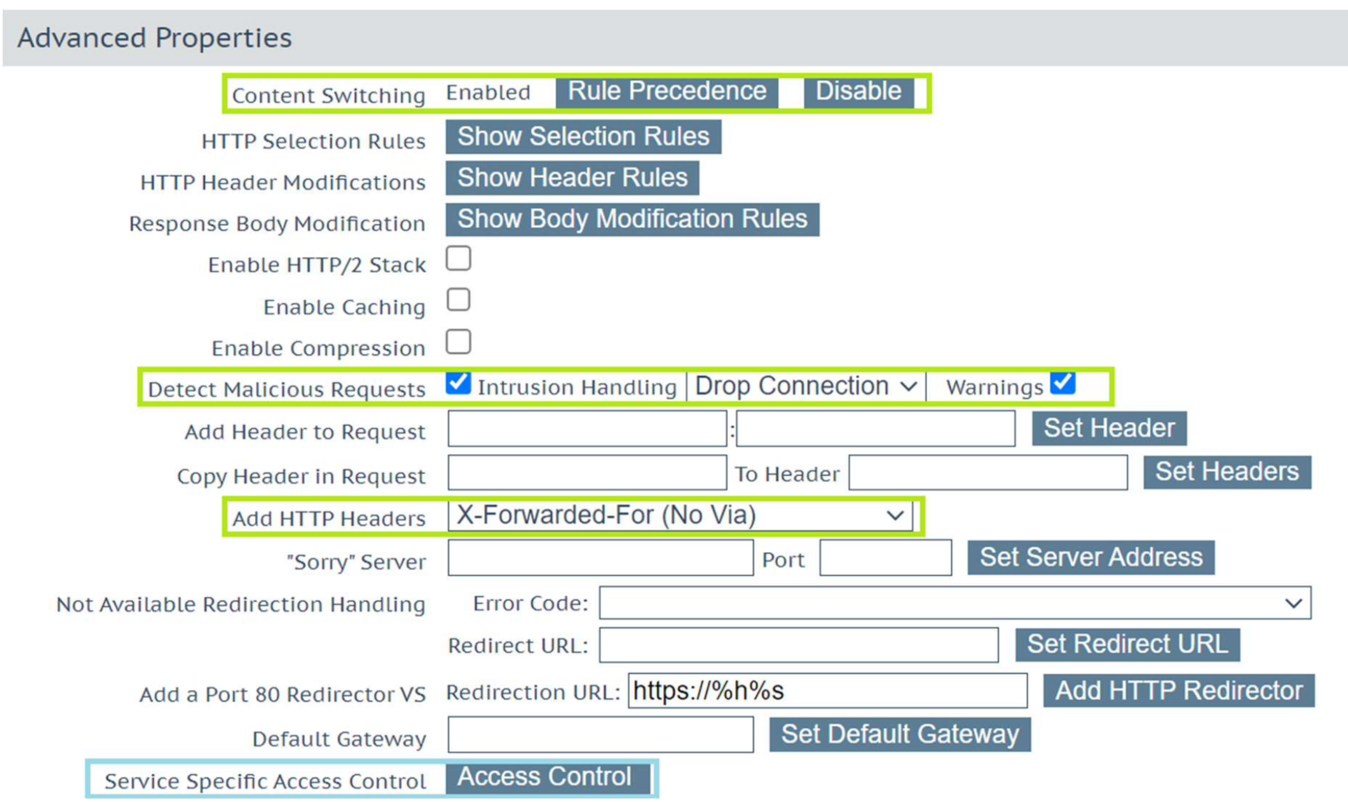

Fig. 5. We enabled IDS/IPS in LoadMaster as an additional layer of defence for little configuration.

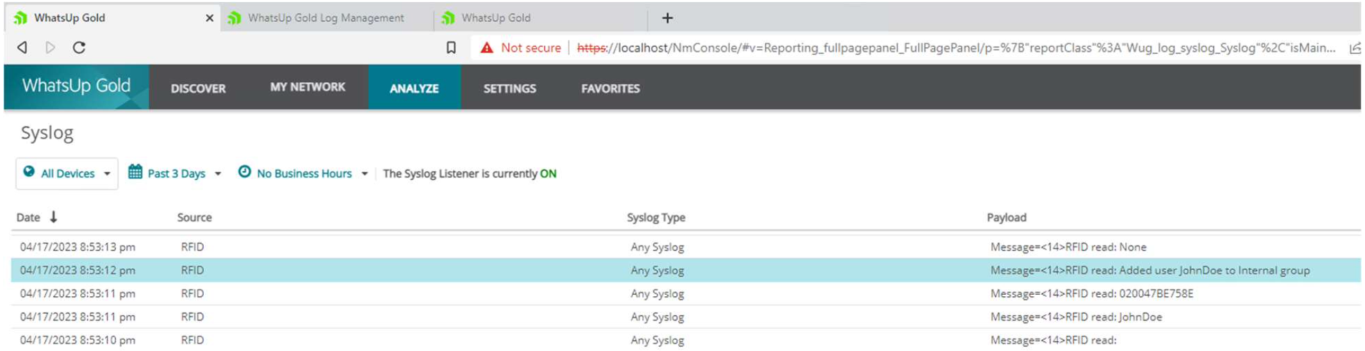

Fig. 6. ...

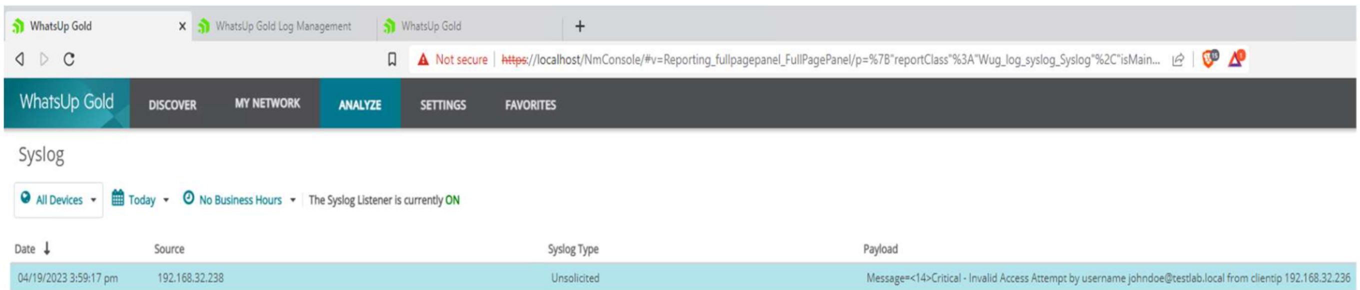

Fig. 7. ...

4) The DNS points to the LM for DNS resolution of the Web resource and since it is an internal request the user is sent to the internal service.
5) The ESP requires the user to login.
6) The user's group membership is checked by the ESP SSO system and they are connected to the appropriate Web resource.
7) An external threat actor attempts to access the Web resource using John Doe's credentials.
8) The DNS points the threat actor to the LM for DNS resolution.
9) Using the correct credentials, the external threat actor logs in via the ESP SSO page.
10) The group membership is read as Internal, but the source is external, so the threat actor is directed to a "server unavailable" page and their IP address is logged to the SIEM service (see Fig. 7).

B. *External User with Internal Threat*

In the second case we are concerned with internal threats: rather than credentials being leaked externally, they are accessed by a threat internally, e.g., someone who may have physical access to the user's desk. The user, Jane Doe, leaves the office for lunch and swipes her access card on the RFID reader on the way out. Her group membership is set to External. Meanwhile an internal threat actor attempts to login from inside the office while the user is away. They are denied access and an event is logged to the SIEM service (WUG) as a breach attempt. The timeline of events is as follows:

1) Jane Doe leaves her office and swipes his access card.
2) The user's group membership is changed from Internal to External. This event is logged to the SIEM service from the Raspberry Pi.
3) An internal threat actor attempts to access the Web resource using Jane Doe's credentials.
4) The DNS points the threat actor to the LM for DNS resolution.
5) Using the correct credentials, the internal threat actor logs in via the ESP SSO page.
6) The group membership is read as External, but the source is internal, so the threat actor is directed to a "server unavailable" page. The event is logged to the SIEM service.

V. Conclusion

References

[1] N. Clarke, *Transparent User Authentication: Biometrics, RFID and Behavioural Profiling*, 1st ed. Springer, 2011.
[2] D. C. Ranasinghe and P. H. Cole, "Confronting security and privacy threats in modern RFID," in *The Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 2058–2064.
[3] T. Kriplean, E. Welbourne, N. Khoussainova, V. Rastogi, M. Balazinska, G. Borriello, T. Kohno, and D. Suciu, "Physical access control for captured RFID data," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 48–55, 2007.

[4] G. Ostojić, S. Stankovski, and M. V. Jovanovic, “Implementation of RFID technology in parking lot access control system,” in *Annual RFID Eurasia*, 2007, pp. 1–5.
[5] U. Farooq, M. ul Hasan, M. Amar, A. Hanif, and M. U. Asad, “RFID based security and access control system,” *IACSIT International Journal of Engineering and Technology*, vol. 6, no. 4, pp. 309–314, 2014.
[6] A. Larchikov, S. Panasenko, A. V. Pimenov, and P. Timofeev, “Combining RFID-based physical access control systems with digital signature systems to increase their security,” in *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014, pp. 100–103.
[7] R. M. Woo-Garcia, anbd U. H. Lomeli-Dorantes, F. López-Huerta, A. L. Herrera-May, and J. Martínez-Castillo, “Design and implementation of a system access control by RFID,” in *IEEE International Engineering Summit (IE-Summit)*, 2016, pp. 1–4.
[8] E. Uzun, V. Atluri, S. Sural, J. Vaidya, G. Parlato, A. L. Ferrara, and M. Parthasarathy, “Analyzing temporal role based access control models,” in *ACM Symposium on Access Control Models and Technologies*, 2012, pp. 177–186.
[9] J. Luo, H. Wang, X. Gong, and T. Li, “A novel role-based access control model in cloud environments,” vol. 9, no. 1, pp. 1–9, 2016.
[10] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, “Detecting and preventing cyber insider threats: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.
[11] A. Chatterjee, Y. Pitroda, and M. Parmar, “Dynamic role-based access control for decentralized applications,” in *International Conference on Blockchain*. Springer, 2020, pp. 185–197.
[12] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420, 2020.
[13] E. Bertino, “Zero trust architecture: Does it help?” *IEEE Security & Privacy*, vol. 19, no. 5, pp. 95–96, 2021.
[14] J. Garbis and J. W. Chapman, *Zero Trust Security*, 1st ed. Apress, 2021.
[15] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid, “Developing cyber-resilient systems: A systems security engineering approach,” https://csrc.nist.gov/pubs/sp/800/160/v2/r1/final, 2021.
[16] Q. Yao, Q. Wang, X. Zhang, and J. Fei, “Dynamic role-based access control for decentralized applications,” in *International Conference on Control, Robotics and Intelligent System*. ACM, 2021, pp. 123–127.
[17] L. Meng, D. Huang, J. An, and X. Zhou, “A continuous authentication protocol without trust authority for zero trust architecture,” *China Communications*, vol. 19, no. 8, pp. 198–213, 2022.
[18] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, “Zero trust architecture (ZTA): A comprehensive survey,” *IEEE Access*, vol. 10, pp. 57 143–57 179, 2022.
[19] Progress Kemp, “LoadMaster,” https://kemptechnologies.com/.
[20] ——, “WhatsUp Gold,” https://kemptechnologies.com/.
[21] Cisco, “Snort,” https://www.snort.org/.
[22] MaxMind, “GeoIP2 Databases,” https://www.maxmind.com/en/solutions/ip-geolocation-databases-api-services.