

MHN Test Dökümanı

MHN(Modern Honey Network) honeypotları merkezi olarak yönetim ve data izleme aracıdır. Bu dökümanda MHN ve honeypot araçlarının test yöntemleri gösterilmiştir.

1. MHN

MHN kullanım dökümanındaki gibi kurulduktan sonra doğru bir şekilde aşağıdaki sayfalar tarayıcı tarafından açılarak merkezi cihazın hazır olduğu teyit edilir.

Merkezi cihazın IP adresinin 169.254.1.1 olduğunu varsayarsak.

<http://169.254.1.1>

<http://169.254.1.1:3000> (Atak haritası)

İlk adresten ‘Deploy’ sekmesine girilerek honeypot yapılandırmalarında kullanılacak deploy kodu alınır. Bu kod her honeypot container için aynı olacaktır. MHN cihazı yeniden yüklenirse değişir.

Kurulan her bir honeypot ‘Sensors’ sekmesinden gözükmelidir. Oluşan ataklar ‘Attacks’ sekmesinde görülür.

2. Cowrie Honeypot

Cowrie, ssh ve telnet tuzak aracıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında cowrie container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n cowrie

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#supervisorctl status cowrie

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için containera belirlenen porttan ssh veya telnet bağlantısı kurulur.

#ssh root@10.0.0.4

#telnet 10.0.0.4 2223

3. Dionaea Honeypot

Dionaea, bir çok protokolden honeypot sağlar. Fakat ahtapot projesinde http ve https protocoelleri için kullanılmıştır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında dionaea container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n dionaea

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#supervisorctl status dionaea

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için container IP adresine tarayıcı ile http ve https bağlantı atılır.

http://10.0.0.4

https://10.0.0.4

4. FTP Honeypot

Bu playbook FTP ve FTPS protokolleri için honeypot yaratır. Honeypot-ftp aracı kullanılmıştır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında ftp container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n ftp

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status honeypot-ftp

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için containera FTP protokolü ile bağlanılır. Filezilla kullanılabilir.

5. POP3 Honeypot

Bu playbook POP3 protokolleri için honeypot yaratır. Honeypot-pop3 aracı kullanılmıştır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında pop3 container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n pop3

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status honeypot-pop3

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için containerın pop3 portuna istek atılır.

#telnet 10.0.0.4 25

#telnet 10.0.0.4 465

6. Amun Honeypot

Amun zaafiyet içeren tuzak aracıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında amun container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n amun

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status amun

komutu ile aracın çalıştığı görülür.

Amun birçok portu dinler. Test etmek için aşağıdaki gibi bağlantı atılabilir.

#telnet 10.0.0.4 25

#telnet 10.0.0.4 80

#telnet 10.0.0.4 21

7. Elasticsearch Honey Pot

Bu araç elasticsearch uygulamasının tuzağıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında elasticsearch container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n elasticsearch

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status elasticsearch

komutu ile aracın çalıştığı görülür.

Aşağıdaki komut ile loglama testi yapılabilir.

#curl http://10.0.0.4:9200/_search?pretty -XPOST -d '{"script_fields": {"myscript": {"script": "java.lang.Math.class.forName(\"java.lang.Runtime\")"}}}'

8. Glaspot Honey Pot

Glaspot web uygulama tuzak aracıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında glaspot container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n glaspot

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#supervisorctl status glaspot

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için honeypot a web istekleri atılır.

http://10.0.0.4

9. P0f honeypot

P0f pasif parmak izi tanımlayıcıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında p0f container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n p0f

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#supervisorctl status p0f

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için container tcp bağlantı atılması yeterlidir.

10. Shockpot Honeypot

Shockpot shellsock zafiyeti için tuzak aracıdır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında shockpot container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n shockpot

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#supervisorctl status shockpot

komutu ile aracın çalıştığı görülür.

Loglama testi yapmak için aşağıdaki şekilde bağlantı atılır

#curl -A '() { ;; }; /bin/lx' http://10.0.0.4

11. SMTP Honeypot

Bu playbook SMTP protokolü için honeypot yaratır. Honeypot-smtp aracı kullanılmıştır. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında smtp container 1 oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n smtp

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status honeypot-smtp

komutu ile aracın çalıştığı görülür. Loglama testi için aşağıdaki gibi bağlantı atılabilir.

#telnet 10.0.0.4 25

#telnet 10.0.0.4 465

12. Suricata Honeypot

Bu playbook MHN ile entegre edilmiş suricata kurar. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında suricata container 1 oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n suricata

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status suricata-rup.service

#systemctl status suricata-rup.timer

komutu ile aracın çalıştığı görülür. Test etmek için cihaz üzerinde aşağıdaki komut çalıştırılabilir.

#apt update

13. Wordpot Honeypot

Wordpot wordpress tuzak sistemidir. Kullanım dökümanındaki gibi kurulum yapıldıktan sonra host cihazında wordpot container ı oluştuğundan emin olunur.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n wordpot

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status wordpot

komutu ile aracın çalıştığı görülür. Test etmek için wordpress admin sayfasına bağlanılır.

http://10.0.0.4/wp-admin

14. Conpot Honeypot

Conpot bir simulasyon aracıdır. Default profil Siemens S7-200 CPU i simule eder. Bu simulasyon modbus, http, snmp ve s7comm protokollerini içerir.

#lxc-ls

komutu varolan containerları listeler.

#lxc-attach -n conpot

komutu ile containera bağlanılarak ayakta olduğu teyit edilir. Daha sonra

#systemctl status conpot

komutu ile aracın çalıştığı görülür. Test etmek için http ya da telnet istegi atılabilir.

http://10.0.0.4:8800

#telnet 10.0.0.4 5020