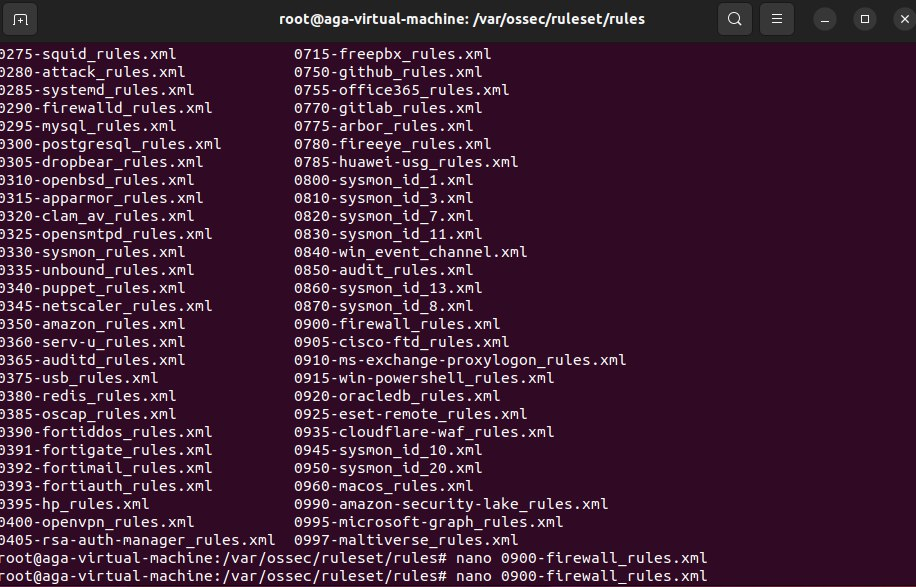
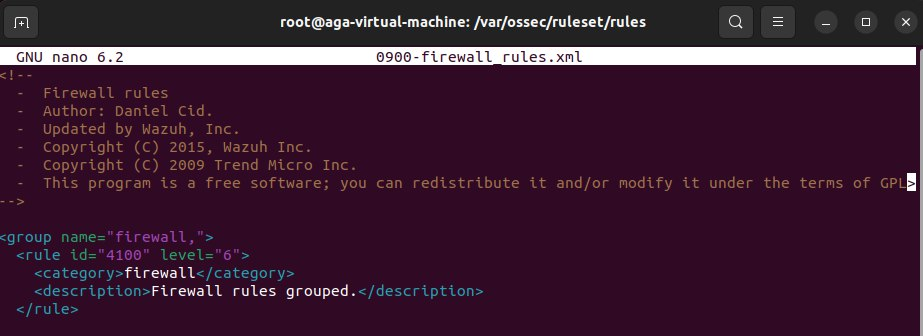
Nmap taraması yakalanışı

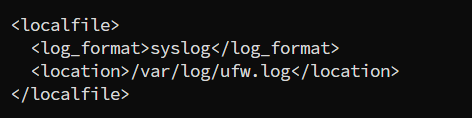
İlk olarak wazuhun içindeki ruleslara bakıyoruz.



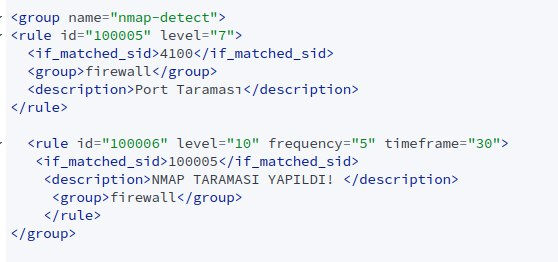
Firewall kısmında ufak bir ayar lazım burada.

Buradaki leveli sıfır değil 6 yapıyoruz.

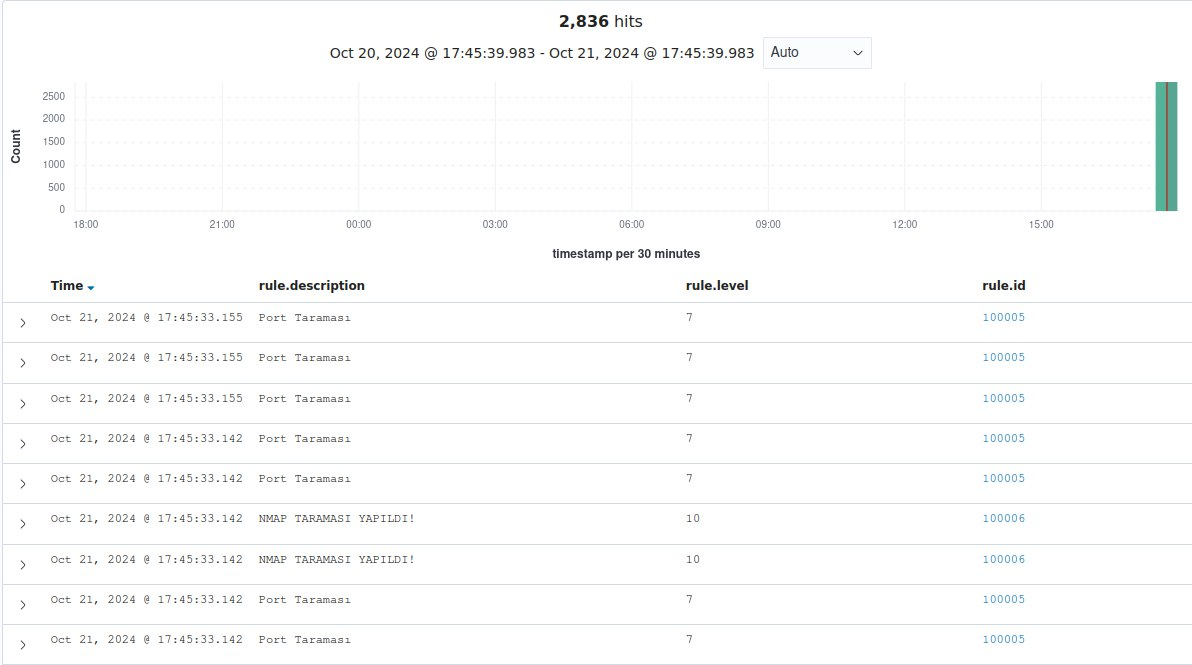
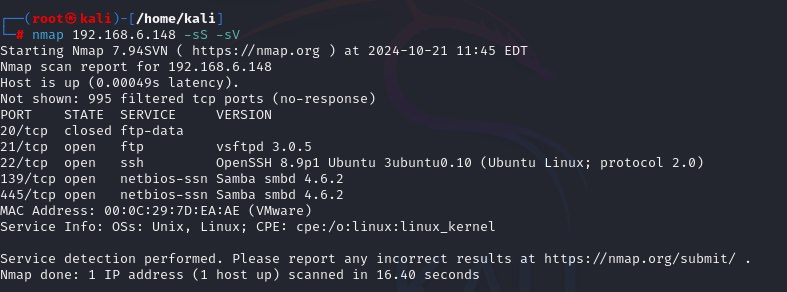
Sonra ossec.conf ufw.log ekliyoruz.



Kural kısmında ise



Geriye kalan kuralı tetiklemek.



Ödev 3 -

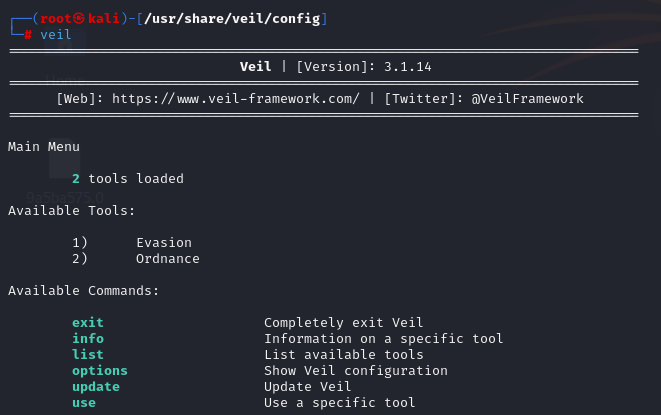
Windows bilgisayardan meterpreter ile shell alındıktan sonra RDP ile bilgisayara erişilmesi

İlk olarak Veil indirerek işlemimize başlıyoruz. Veil de eğitimde gördüğünüz FatRat gibi bir virüs yaratıcısı. FatRat en son kullandığımda çalışmıyordu direkt Veil’i indirdim. İndirmesi bir tık uzun sürüyor. Evinizde güvenli internet varsa veya okul internetinden indirmeye çalışıyorsanız indirmeyebilir. Komut sistemine Veil yazarsanız direkt indireyim mi diye soruyor.

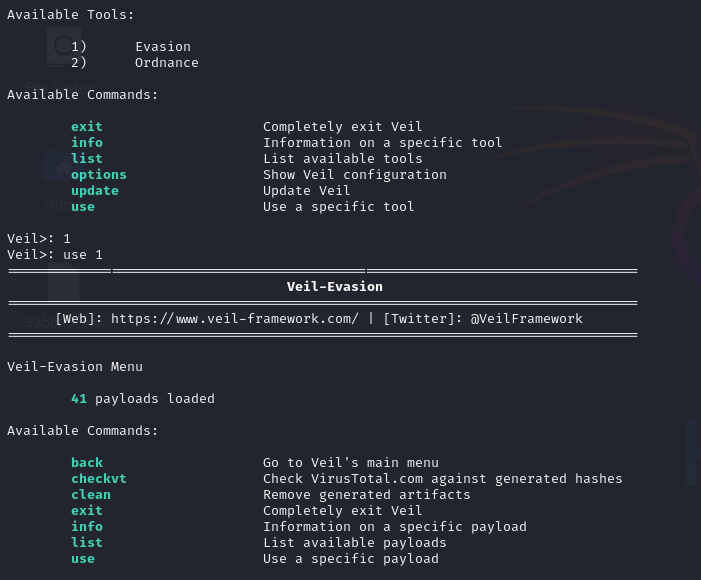
Veil’e göz atalım. Kurulum aşaması böyle başlıyor.



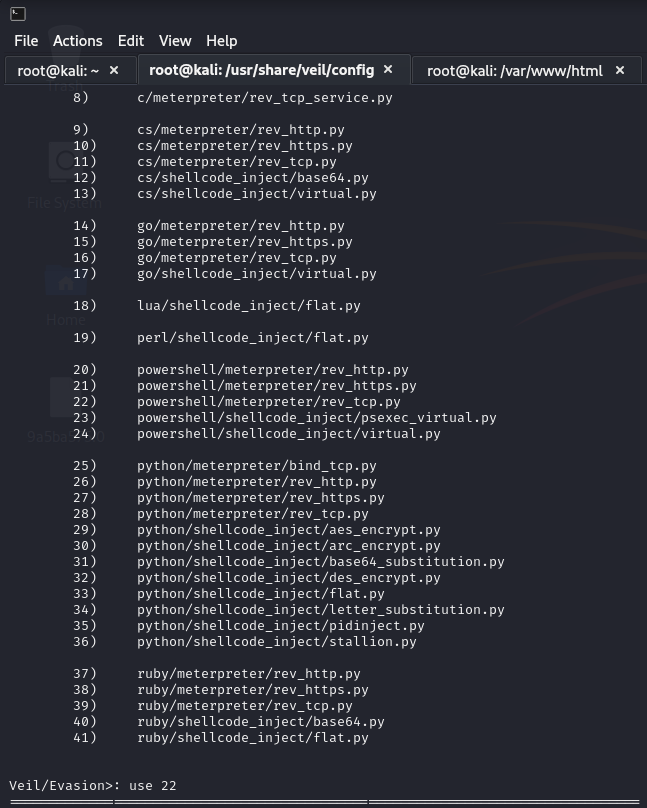
İlk ekranımız böyle başlıyor.

Evasion modülü, oluşturulan zararlı payloadın antivirus ve benzeri güvenlik çözümlerinden kaçmasına yardımcı olan teknikleri içerir. Ordnance modülü, çeşitli saldırı vektörleriyle hedef sistemlere saldırmak için kullanılan farklı saldırı türlerini içerir.

Biz Evasion kullanacağız use 1 diyerek

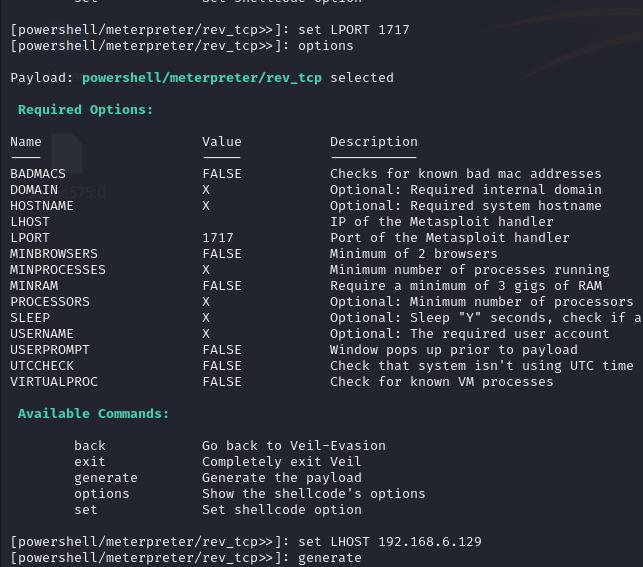


Sonraki adımda list diyerek payloadımızı belirlememiz lazım. Use 22 diyerek meterpreter payloadını seçiyoruz.



Sonraki ekranda ise LPORT ve LHOST belirleyeceğiz. LPORT’u unutmayın. İlerde kullanacağız. Ben LPORTumu 1717 yaptım ve LHOSTa kendi IPmi girdim localhost biz olacağız. Diğer ayarların açıklaması yanlarında yazıyor. Bunları kullanarak yakalanma olasılığını azaltabiliriz fakat Windows Defender basit virüsleri artık çok kolay yakalıyor.

Son olarak generate diyip virüsümüzü yaratıyoruz ismini veriyoruz

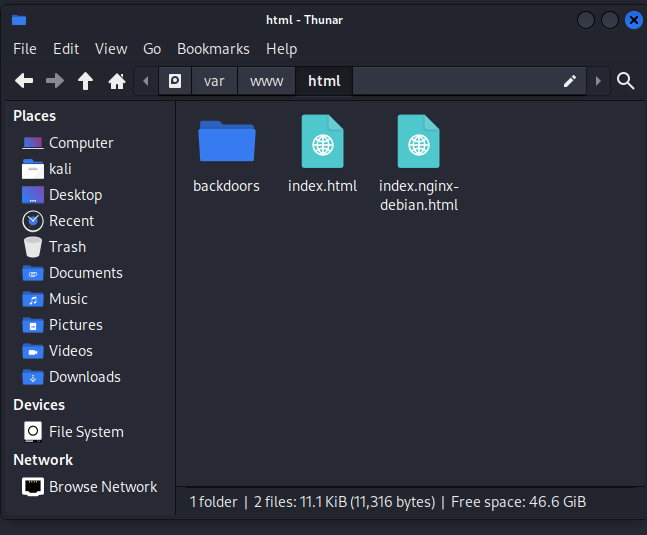


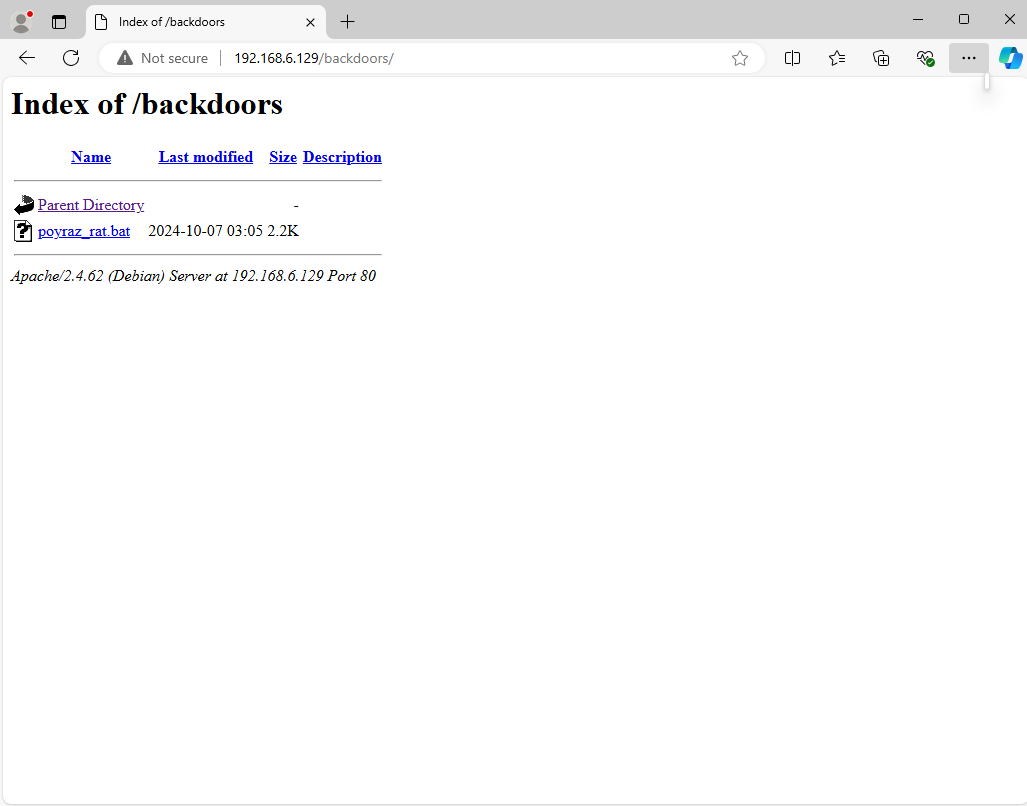
Son ekran böyle geliyor ve dosyamızın bulunduğu yeri gösteriyor.

Windows makinamıza bunu geçirmek için ben apache2 servisini kullandım.

Service apache2 start diyorsunuz.

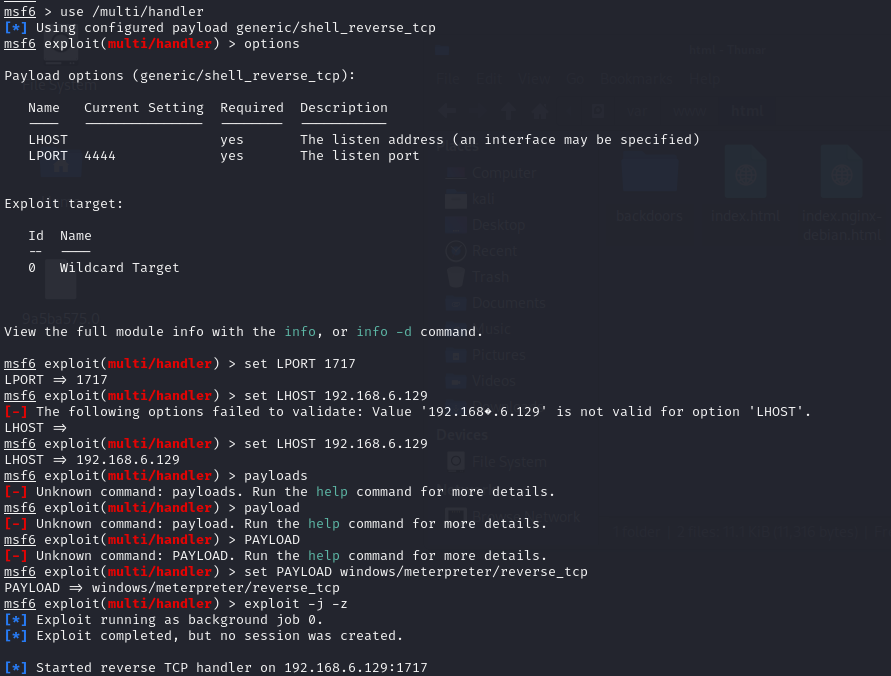
IP adresini yazarak girdiğiniz apache serverın index sayfası burada bulunuyor buraya bir dosya oluşturuyorum ve dosyayı içine atıyorum.





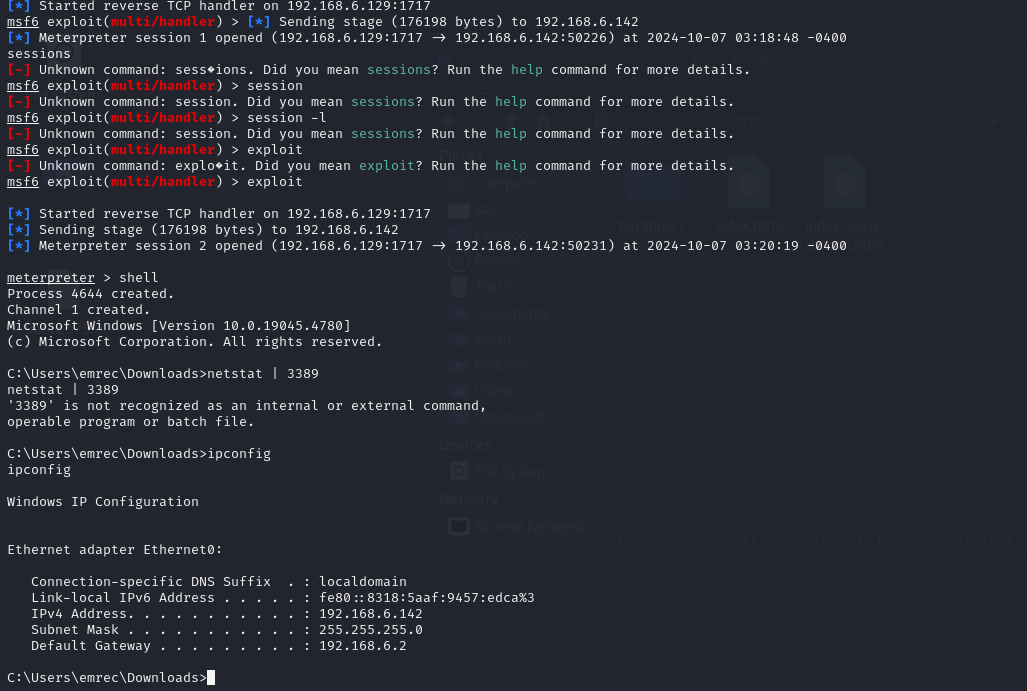
Dosyayı bu sayfadan indiriyorum. Defenderı kapatmayı unutmayın indirirken bile virüs bu indirme bunu uyarısı geliyor.

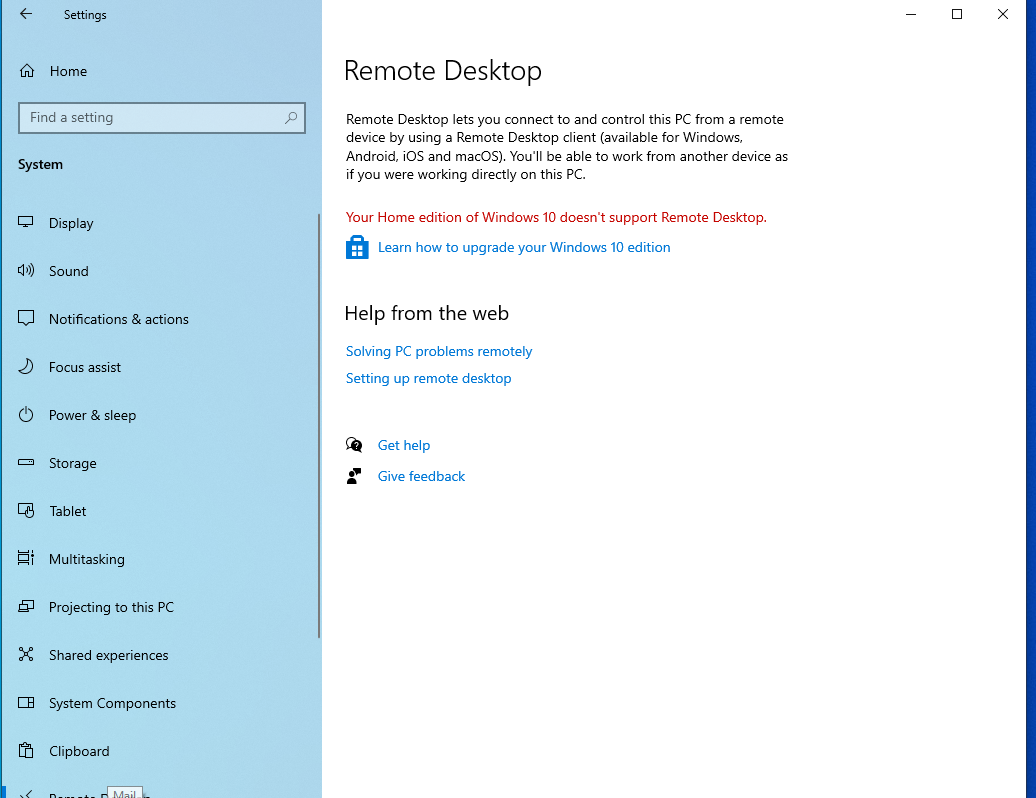
Şimdi msfconsole açıp /multi/handler modülüne gidelim.

Modüle geldikten sonra LPORT ve LHOST ayarlayın. Size port numaranızı unutmayın demiştim. Normalde payloads yazdığınızda tüm payloadlar geliyordu burda olmadı ama payloadı internet buldum ve ayarladım. Veil ve msfconsole için reverse\_tcp seçtim çünkü bağlantının bana karşıdan gelmesini istiyorum. Makinem dinlemeye başladı.

Windowsuma gidip dosyamı çalıştırıp session gelmesini bekliyorum.

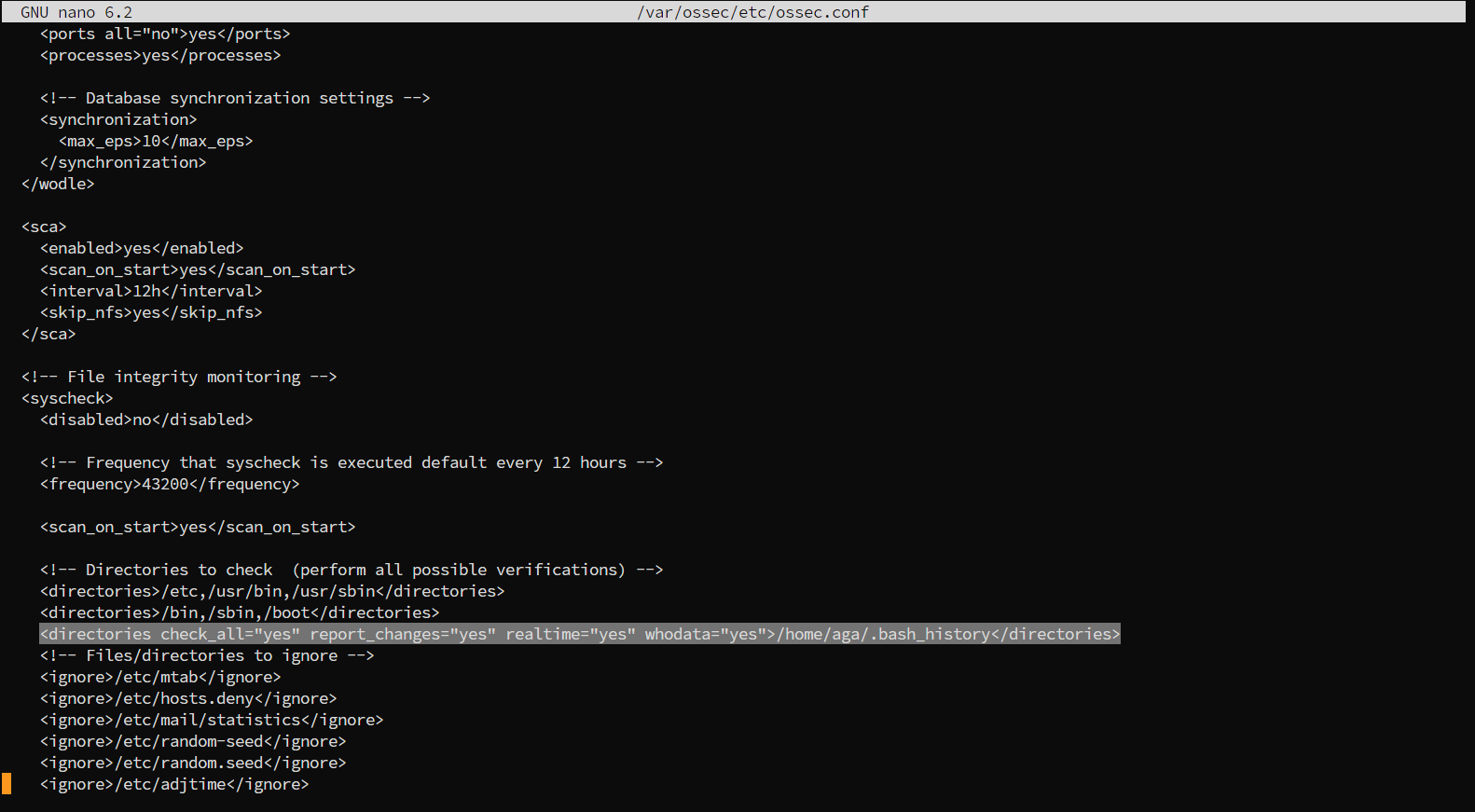
(Ekstra bir dip not: NGROK kullanarak aldığınız IPyi tünelleme için kullanabilirsiniz. Bu daha da güvenli bir yol sağlar. Bunun için Veil’in python kodunda biraz değişiklik yapmanız gerekiyor.)

Yanlış yazdığınızda size yardımcı oluyor program. Meterpreter geldiğinde shell yazıp komut istemcisine gidebiliyorsunuz.

Windowsum ne hikmetse RDP desteklemiyorum ben dedi ve burda bitti. Ekstra bir sanal makina için internet sıkıntı oluyor. KYK sağolsun.

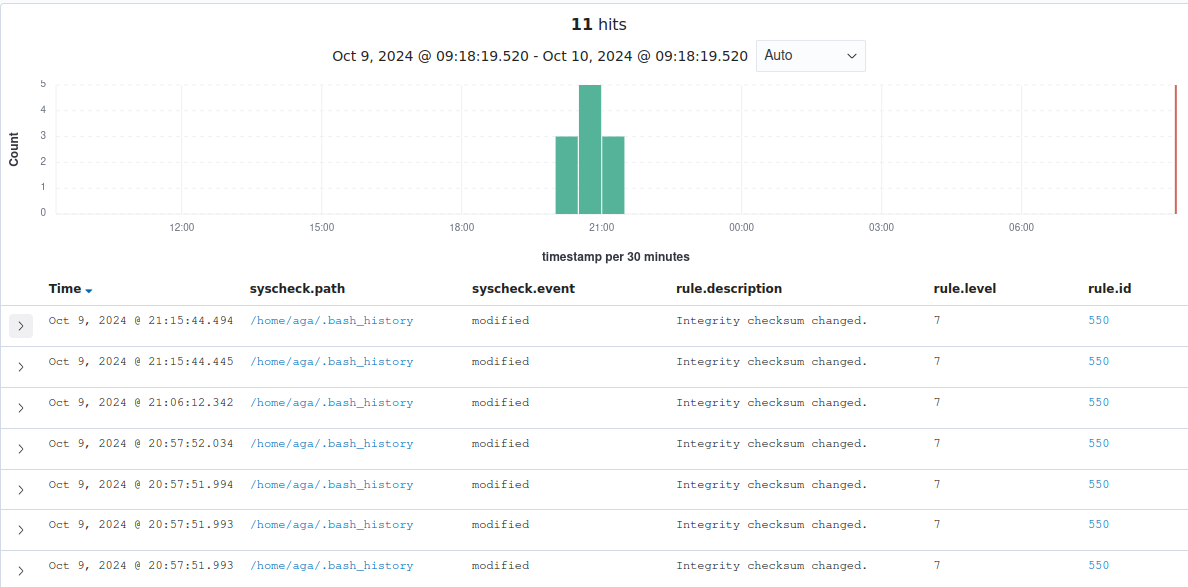
ÖDEV-2

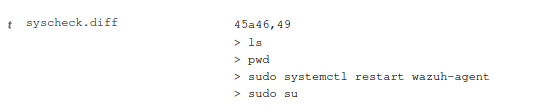
Ubuntu serverda .bash\_history değişikliklerini kontrol etmek istiyoruz bunun için ossec.conf dosyasına gidip syscheck’e bir kod eklememiz lazım.



* **<directories>**: Bu etiket, hangi dizinlerin izleneceğini tanımlar. Burada /home/aga/.bash\_history dizini izlenmektedir.
* **check\_all="yes"**: Bu seçenek, Wazuh’un tüm dosyaları kontrol etmesini sağlar. Yani, izlenen dizindeki dosyaların içerikleri ve özellikleri tamamen kontrol edilecektir.
* **report\_changes="yes"**: Bu seçenek, herhangi bir değişiklik tespit edildiğinde rapor oluşturulmasını sağlar. Eğer dosyalarda bir değişiklik olursa, bu değişiklikler raporlanır.
* **realtime="yes"**: Gerçek zamanlı izleme anlamına gelir. Dosya veya dizinlerde herhangi bir değişiklik olduğunda, Wazuh bu değişikliği anında tespit eder ve gerekli uyarıları üretir.
* **whodata="yes"**: Bu seçenek etkinleştirildiğinde, dosya ve dizinlerde yapılan değişikliklerin kim tarafından yapıldığı da kaydedilir. Yani, dosya değişikliğine yol açan kullanıcı, işlem kimliği gibi bilgiler toplanır

Aga kullanıcısı terminalde komut çalıştırdığında düşmesi lazım şimdi. Değişikliğin ne olduğunu gizlemek için <no-diff> içine de alabilirsiniz.





İlk baştaki içeriğe bakarsanız eğer syscheck.diff içinde değişiklikleri bulabilirsiniz.

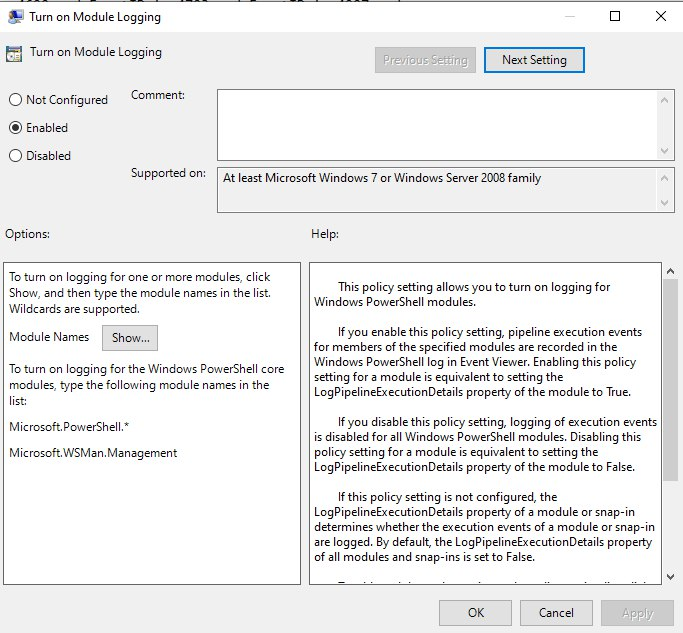
Bunun sıkıntısı hacker girip yeni kullanıcı oluşturursa hiçbir log elde edilmiyor.

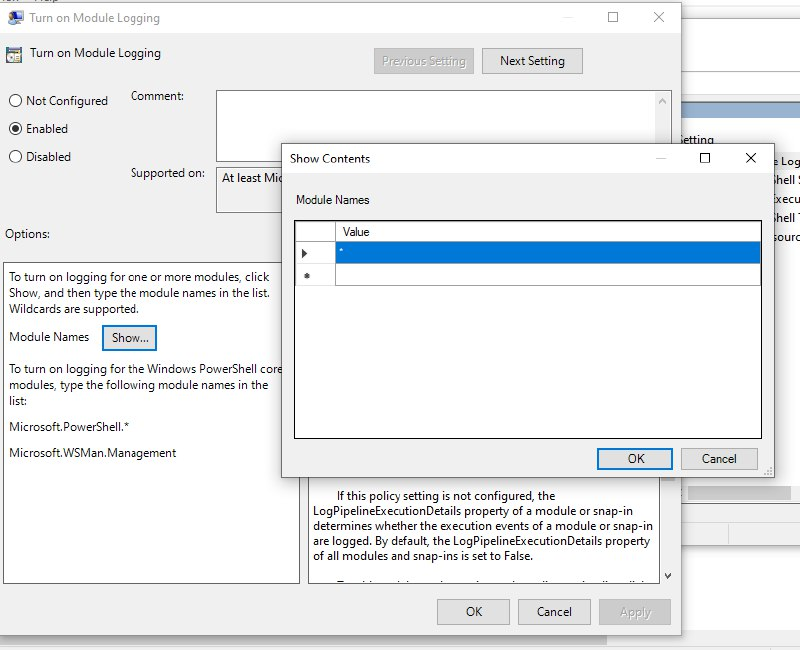
POWERSHELL LOG

Bu ödevi iki farklı yolla yaptım ama ikinci yolun biraz sıkıntıları var ilk olarak full halini atacağım.

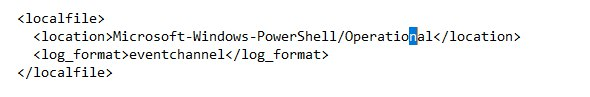
Öncelikle grup ilkelerinden Module Logging PowerShell transcription ve PowerShell script modül loggingi açmamız lazım. Ekstra olarak modül logging kısmında modül names kısmında \* eklememiz gerekiyor. Bu tüm logları al anlamına geliyor.

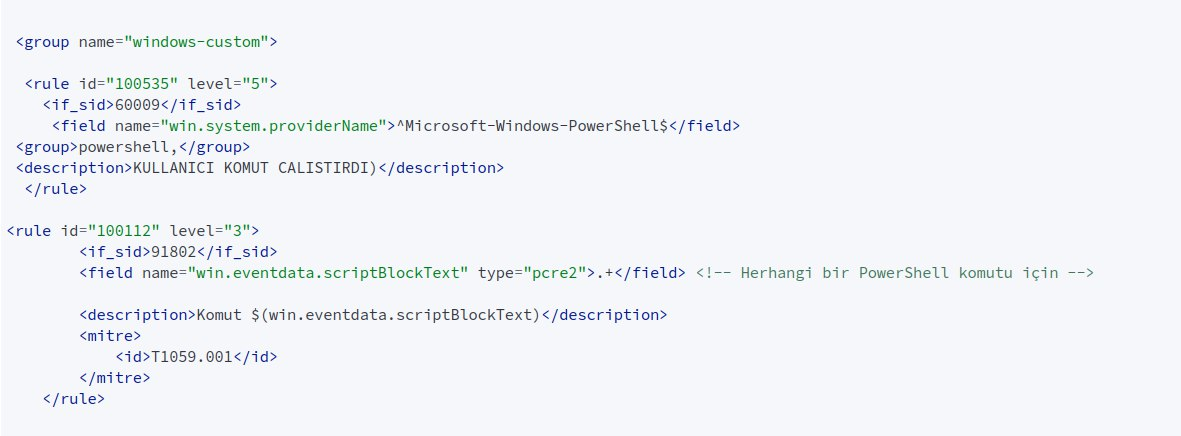




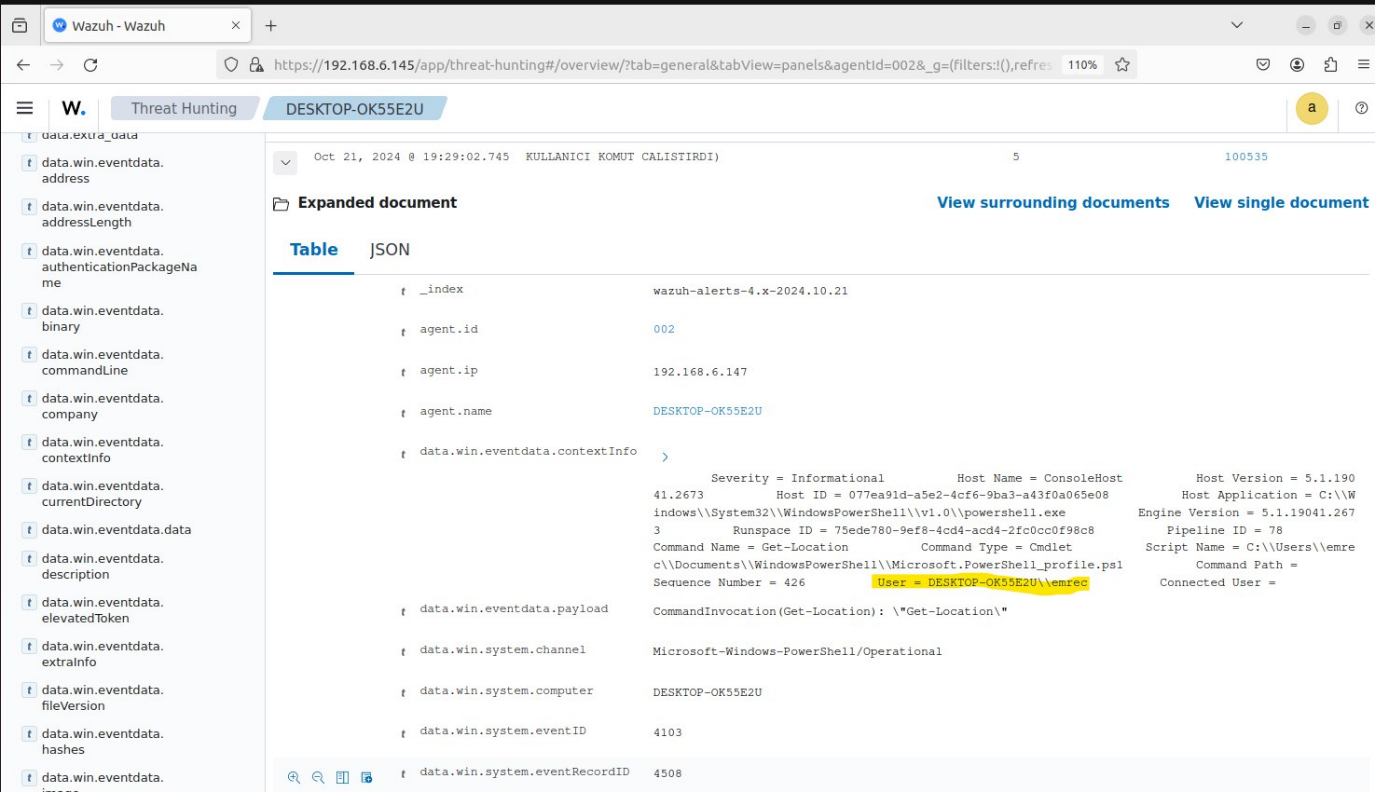
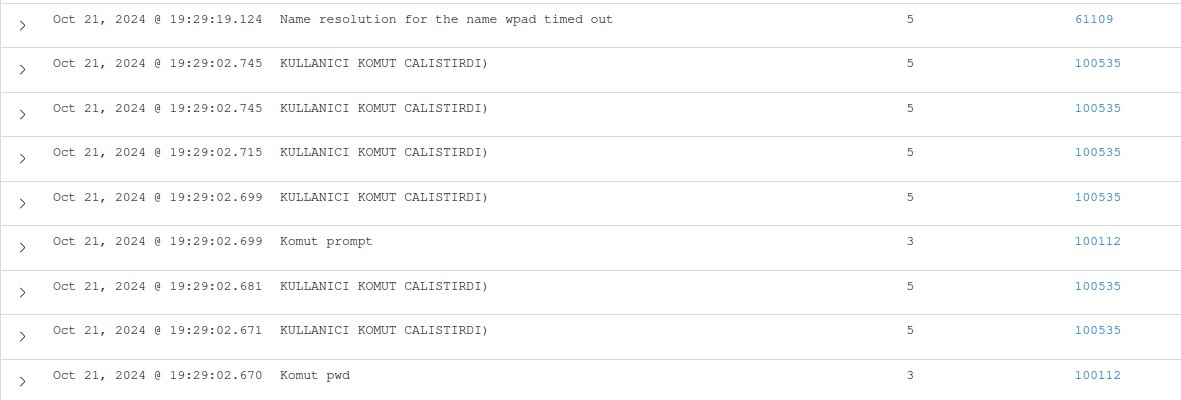


Event logları almak için ossec.conf altına şunu da eklemelisiniz.



Alttaki kuralı yazarken powershell rulesın içinde bazı komutlar için log düştüğünü gördüm. Bir kural için çalışıyorsa tüm komutlar için de çalışabilir diye düşünerek gptden yardım alarak kuralı son haline getirdim. İlk kural ise powershellde yazılan herhangi bir şey log olarak düşüyordu ve tekte 8 log düşüyordu bunun içinden komutu bulmak zor olacaktı. İlk kuralın iyi yanı çalıştıran kullanıcıyı göstermesiydi. İkisini birden gösteren tek kuralda uğraştım ama çözüm bulamadım ben de ayrı ayrı çalışmasına karar verdim.

Pwd yazdığımızda olanlara bakalım.



İkinci yoldan kısaca bahsedeceğim. Kısaca keseceğim oluşturduğumuz bir .txt yolu ile file integrity monitoring yaparak yakalayabiliyoruz. Burda sıkıntı ise yazan kişi kim bilmiyoruz ve real time yakaladığı için loglarda yazılan komutlar harf harf gözüküyor.

Aşağıdaki kodu notepad.exe $profile diyerek powershellde açıyoruz.

$inputLogPath = "C:\Users\emrec\inputCommands.txt"

function prompt {

$lastCommand = $(Get-History -Count 1 | Select-Object -Last 1).CommandLine

if ($lastCommand) {

$lastCommand | Out-File -Append $inputLogPath

}

return "PS " + (Get-Location) + "> "

}

# Profil dosyasına prompt fonksiyonunu ekleyelim

Add-Content $profile -Value @'

function prompt {

$lastCommand = $(Get-History -Count 1 | Select-Object -Last 1).CommandLine

if ($lastCommand) {

$lastCommand | Out-File -Append $inputLogPath

}

return "PS " + (Get-Location) + "> "

}

'@

function prompt {

$lastCommand = $(Get-History -Count 1 | Select-Object -Last 1).CommandLine

if ($lastCommand) {

$lastCommand | Out-File -Append $inputLogPath

}

return "PS " + (Get-Location) + "> "

}

function prompt {

$lastCommand = $(Get-History -Count 1 | Select-Object -Last 1).CommandLine

if ($lastCommand) {

$lastCommand | Out-File -Append $inputLogPath

}

return "PS " + (Get-Location) + "> "

}

function prompt {

$lastCommand = $(Get-History -Count 1 | Select-Object -Last 1).CommandLine

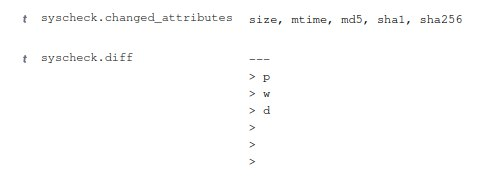
if ($lastCommand) {

$lastCommand | Out-File -Append $inputLogPath

}

return "PS " + (Get-Location) + "> "

}



Bu da biraz komik duruyor.