Ücretsiz ve Açık Kaynak Kodlu Araçlar Kullanılarak Geliştirilen Ağ Güvenlik Açığı Tarayıcısı

Savaş Saygılı[1] Murat Koyuncu[2]

[1]Finansal Teknoloji Hizmetleri A.Ş., Ankara
[2]Bilişim Sistemleri Mühendisliği Bölümü, Atılım Üniversitesi, Ankara

[1]e-posta: sasaygili@fintek.com.tr [2]mkoyuncu@atilim.edu.tr

Özetçe

Bu çalışma kapsamında internet üzerinde bulunan ücretsiz ve açık kaynak kodlu bir çok araç bir araya getirilerek yerel alan ağı güvenlik testleri yapan güçlü bir zafiyet tarama aracı geliştirilmiştir. Geliştirilen kullanıcı dostu arayüz ve raporlama modülleriyle mevcut güvenlik araçlarının çıktıları daha anlaşılabilir hale getirilmiştir. Geliştirilen araç sayesinde, ağların güvenlik zafiyet testleri daha kolay yapılabilmektedir. Böyle bir aracın geliştirilmesindeki amaç, ağ veya güvenlik yöneticilerinin güvenlikle ilgili ürün ve destek sağlayan firmalara çok para ödemeden sistemlerini güvenlik tehditlerine karşı test etmelerine ve açıklarını kapatmalarına yardımcı olmaktır.

1. Giriş

Günümüzde internet kuruluşların ve kişilerin iş yapma yöntemleriyle ilgili büyük değişiklikler getirmiş ve iş yapma etkinliğini artırmıştır. İnternet olmadan onun sağlamış olduğu etkinliği ve iletişimi başka türlü yakalamak mümkün olmamaktadır. İnternet tartışmasız bir şekilde hayatın parçası olmuştur. Diğer taraftan internet, bilgisayar korsanları gibi kötü amaçlı çalışan kişilerce yapılan, kuruluşlarda etkinliği düşüren ve kaynak israfına yol açan bir takım sorunları da beraberinde getirmiştir. Bu nedenle, kuruluşların ağlarını ve bilgisayarlarını saldırılara karşı güvenli hale getirerek, görevlerini yerine getirirken kesinti yaşanmamasını sağlamaları gerekmektedir [1][2][3][4].

Güvenli bir bilgi sistem ortamı oluşturmak için, ağ ve sistemlerin düzenli olarak test edilmesi, mevcut zafiyetlerin belirlenmesi ve giderilmesi bir zorunluluktur. Burada kastedilen zafiyet, sistemlerde, güvenlik politikalarında veya uygulamalarda mevcut ve güvenlik tehdit kaynaklarınca kullanılabilecek zayıflıklardır [5]. Genellikle bir ağdaki güvenlik açıklarını ortaya çıkarmak için ilk işlem olarak ağ ve sistem özelliklerini ortaya çıkaracak bir zafiyet testi yapılır. Bu test, işletim sistemleri, dosya sistemleri, paylaşımlar, kullanıcı hesapları, kullanıcı hakları, açık olan portlar, yazılım güncellemeleri, güvenlik yamaları, anti-virus programları, güvenlik duvarı gibi birçok verinin kontrol ve analiz edilmesine dayanmaktadır.

Zafiyet testleri yapabilecek araçların olması, güvenlik ve ağ yöneticileri için büyük kolaylıklar sağlamaktadır. Bu tür araçlar sayesinde herhangi bir sorun yaşanmadan açıkların giderilmesi mümkün olabilmektedir. Sistemler başlangıçta güvenlik açısından ne kadar dikkatli kurulursa kurulsun insanlardan veya bilgisayarlarda çalışan yazılımlardan kaynaklanan çeşitli sebeplerle güvenlik açıklarının oluşması her zaman olasıdır. Bu nedenle ağların düzenli olarak kontrol edilmesi çok önemlidir. Bu işlem de ancak düzenli olarak çalıştırılacak araçlarla gerçekleştirilebilir.

Güvenliğin öneminin artmasıyla birlikte mevcut güvenlik test araçlarının sayısı da artmaktadır. Güvenlik açıklarını tespit etmek için NMap, Nessus, GFI LanGuard, SupperScan, SolarWinds, F-Scan, CoreImpact and MaxPatrol gibi birçok araç bulmak mümkündür. En iyi 10 zafiyet tarayıcı yazılım konusunda yapılan bir araştırmanın sonucu [6]'da verilmiştir. Bu araçların birçoğu, özellikle de güçlü olanlar ücretli yazılımlardır. Güvenliğin öneminden ve gizeminden dolayı, birçok kuruluş güvenlikle ilgili araçlar almak ve idame ettirmek için büyük paralar ödemektedir. Diğer taraftan, ücretsiz ve açık kaynak kodlu yazılımlar geliştirme konusunda dünyada büyük bir akım vardır. Güvenlikle ilgili olarak da ücretsiz ve açık kaynak kodlu yazılımlar bulmak ve indirip kullanmak mümkündür. Fakat genel olarak bu yazılımlar, ağ ve bilgisayarların belli özelliklerini test edebilmektedir. Ayrıca, kullanım açısından çok kullanıcı dostu olduklarını ve çıktılarının kolay anlaşılabilir olduğunu söylemek mümkün değildir.

Bu çalışma kapsamında, internet üzerinden indirilebilen ücretsiz ve açık kaynak kodlu araçlar kullanılarak, bir yerel alan ağındaki güvenlikle ilgili parametreleri tespit etmek ve güvenlik açıklarını ortaya çıkarmak üzere güçlü bir Ağ Güvenlik Açığı Tarayıcısı (AGAT) geliştirilmiştir. Geliştirilen araç gerekli yerlerde işletim sistemi komutları ve komut dosyaları (script) ile desteklenmiştir. Farklı işlevleri yapan bir çok güvenlik aracı, komut dosyası ve komut uygun bir

şekilde bir araya getirilerek kapsamlı bir zafiyet tarayıcısı ortaya çıkarılmıştır. Geliştirilen bir arayüzle belirli testlerin veya tüm testlerin otomatik olarak yapılması sağlanmıştır. Ağ veya güvenlik yöneticileri geliştirilen tarayıcıyı çalıştırarak ağdan güvenlikle ilgili önemli olabilecek bilgileri otomatik olarak toplayabilmektedir. Toplanan bilgiler ağdaki açıkların kapatılması için doğrudan kullanılabileceği gibi daha ayrıntılı testler yapmak için ön bilgi olarak da kullanılabilmektedir.

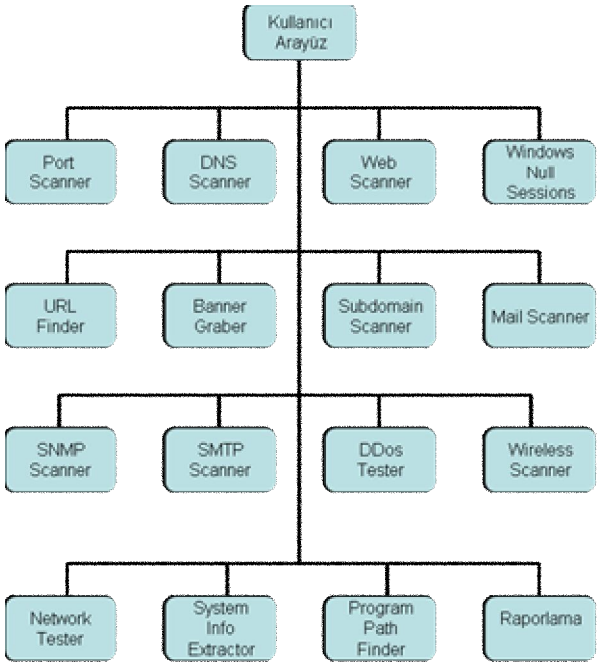
Bu çalışmanın özgün yönleri şunlardır: 1) Ücretsiz ve açık kaynak kodlu yazılımlar kullanılarak, mevcutlara göre daha kapsamlı bir güvenlik tarayıcısı geliştirilmiştir. Ağla ilgili daha çok özellik test edilebilmektedir. 2) Arayüz vasıtasıyla kullanıcılara ulaştırılan sonuçlar daha anlaşılır bir hale getirilmiştir. Mümkün olan yerlere açığı kapatmaya yönelik öneriler konulmuştur. 3) Gün geçtikçe açık kaynak kodlu yazılımların kullanılması önem kazanmaktadır. Bu çalışma güvenlikle ilgili olarak açık kaynak kodlu araçlarla neler yapılabileceğini göstermek açısından önemlidir.

Bildirinin bundan sonraki kısmı şu şekilde düzenlenmiştir: 2'nci bölümde geliştirilen sistemin mimarisi verilmiştir. Verilen mimarinin nasıl geliştirildiği 3'ncü bölümde açıklanmıştır. Geliştirilen aracın, mevcutlarla karşılaştırması 4'ncü bölümde verilmiştir. Bildiri sonuç bölümüyle tamamlanmaktadır.

2. Sistem Mimarisi

Sistem mimarisi Şekil-1'de verilmiştir. Tüm sistem toplam 17 modülden oluşmaktadır. Bunlardan birincisi kullanıcı arayüzü, sonuncusu raporlama modülü ve diğer 15 adedi çeşitli test modülleridir. Test modüllerinin her biri, bir veya birden fazla açık kaynak kodlu hazır araçları kullanmaktadır. Ayrıca bazı modüller mevcut işletim sistemi komutlarından veya hazırlanmış komut dosyalarından (script) yararlanmaktadır. İnternet'den indirilerek kullanılan araçların tamamı ücretsiz ve açık kaynak kodlu (free and open source software-FOSS) yazılımlardır [7] [8]. Bu yazılımların tamamı GPL lisanslı yazılımlardır [9]. Komut dosyaları genelde Linux komutlarıyla oluşturulan dosyalardır. Ayrıca Python dili kullanılarak geliştirilen bazı programlarda bu kapsamda değerlendirilmiştir.

Modüllerin içerisinde kullanılan araçlar Tablo 1'de gösterilmiştir. Tablonun birinci sütunu Şekil-1'de gösterilen mimarideki modülleri içermektedir. İkinci sütun her modülün alt modüllerini göstermektedir. Örneğin Port Scanner, Nmap, AMap ve Port Parser olmak üzere üç adet alt modülden (araç/komut/komut dosyası) oluşmaktadır. Üçüncü sütun ise ikinci sütunda verilen aracın türünü göstermektedir. ÜAKK, yazılımın internet üzerinden indirilen ücretsiz ve açık kaynak kodlu bir yazılım olduğunu belirtmektedir. KD, kullanılan yazılımın yazarlar tarafından geliştirilmiş bir komut dosyası (script) olduğunu göstermektedir. K ise kullanılan aracın bir işletim sistemi komutu olduğunu belirtmektedir.



Şekil-1: Sistem Mimarisi

Sistem mimarisi içinde kullanılan ücretsiz ve açık kaynak kodlu araçların önemlileri aşağıda referanslarıyla birlikte verilmiştir.

NMap: Ağ ve bilgisayarlarla ilgili birçok bilgiyi çıkaran oldukça kapsamlı bir araçtır [10].

AMap: Farklı portta çalışsa dahi çalışan ağ uygulamalarını tespit eden bir araçtır [11].

DNSEnum: Alanlar hakkında bilgi toplayan bir araçtır. Kullanılan IP bloklarını ortaya çıkarmaktadır[12].

Nikto: Web sunucularda açıkları tespit etmek üzere kullanılan kapsamlı bir araçtır [13].

DIRB: Web sunucu açıklarını tespit etmek için kullanılmaktadır [14].

SMBClient: Sunuculardaki SMB/CIFS kaynaklarına erişim için kullanılmaktadır [15].

BannerGrab: TCP, UDP ve SSL servislere bağlanmak ve bu servisler hakkında bilgi toplamak için kullanılmaktadır [16].

Subdomainer: Google, MSN Search, Yahoo gibi halka açık kaynakları kullanarak alt alan isimlerini çıkarmaya yarayan bir araçtır [17].

OneSixtyOne: Güçlü bir SNMP tarayıcısıdır [18].
Curl-Loader: Sunucular üzerinde stres testleri yapmak için kullanılmaktadır [19].
Kısmet: Kablosuz ağlar için geliştirilmiş bir test sistemidir [20].
LANMap: Pasif olarak ağdaki trafiği dinleyen ve trafikten ağın bir topolojisini çıkaran programdır [21].

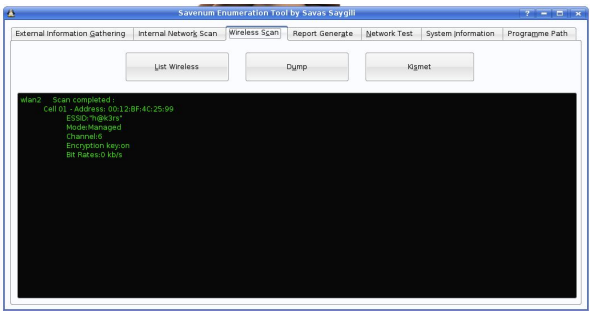
Tablo 1: Modüllerin iç yapısı

Modül	İçerdiği Araçlar	Türü
Port Scanner	Nmap	ÜAKK
	AMap	ÜAKK
	Port Parser	KD
DNS Scanner	Subdomainer	KD
	DNSenum	ÜAKK
	DnsSearch	KD
Web Scanner	Webext	KD
	Nikto	ÜAKK
	DIRB	ÜAKK
Windows Null Sessions	Smbclient	ÜAKK
URL Finder	URLparse	KD
Banner Graber	Bannergrab	ÜAKK
Subdomain Scanner	Subdomainer	ÜAKK
Mail Scanner	Googlemail	KD
SNMP Scanner	OneSixtyOne	ÜAKK
	SNMPEnum	ÜAKK
	SNMPSearch	KD
SMTP Scanner	SMTPVrfy	K
DDos Tester	Curl-loader	ÜAKK
Wireless Scanner	Iwconfig	K
	Kismet	ÜAKK
Network Tester	Ping	K
	Netstat	K
	Mac Changer	ÜAKK
	Tcpdump	ÜAKK
	Lanmap	ÜAKK
System Info Extractor	Ipconfig	K
	Df	K
	Mem	K
	Host	K
	Uname	K
Program Path Finder	Which	K

3. Uygulama

Ağ Güvenlik Açığı Tarayıcısı, birçok farklı aracın bir araya getirilmesiyle oluşturulmuştur. Farklı araçlar bir kullanıcı arayüzünün altında birleştirilmişlerdir. Kullanılan araçların birçoğunun, işletim sistemi komut satırında değişik parametreler belirtilerek çalıştırılması gerekmektedir. Özelikle uzman olmayan kişiler için bu durum çok uygun değildir. Geliştirilen grafiksel arayüz ile kullanıcılar fare tıklarıyla ilgili araçlara erişebilmekte ve çalıştırabilmektedir. Böylece birçok araç için kullanıcı dostu bir arayüz imkanı sağlanmıştır.
Kullanıcı arayüzü, ücretsiz ve açık kaynak kodlu bir araç olan Kommander [22] kullanılarak geliştirilmiştir. Kommander herhangi bir programlama diline ihtiyaç duymadan kullanıcı-bilgisayar arayüzleri geliştirme aracıdır. Grafiksel bir arayüz üzerinde kullanıcı diyalogları kolayca tasarlanabilmektedir. Sürükle-bırak yöntemiyle kod yazmasını bilmeyenler dahi kolayca kullanıcı arayüzleri geliştirebilirler. Geliştirilen arayüz elemanlarına herhangi bir araç kullanılarak yazılan komut dosyası (scripting) ilişkilendirilebilmektedir. Bu yöntemle kullanıcı arayüzü ile altta kullanılan modüller birbirlerine bağlanmışlardır.

Ağ Güvenlik Açığı Tarayıcısı, kullanıcılara ağ üzerinde belirli testleri yapma imkanı verdiği gibi, tüm ağ üzerinde kapsamlı testler yapma imkanı da sağlamaktadır. Örnek olarak Şekil-2'de kablosuz ağ test (wireless module) modülünün ekranı verilmiştir. Burada üç seçenek görülmektedir. Bu ekranı kullanarak kullanıcı belirli kablosuz ağ testlerini yapabilmektedir. Bilirli bir konuyu incelemek yerine tüm ağ üzerinde kapsamlı bir inceleme yapılmak istendiğinde ise ağ tarama seçeneği ile tüm testler sistem tarafından otomatik olarak yapılır. Tüm ağı taramak, ağın büyüklüğüne bağlı olarak oldukça uzun sürecek bir işlem olabilmektedir. Böyle bir durumda sistem çalıştırılıp bırakılabilir ve sonuçlar daha sonra alınabilir. Böylece kullanıcının tek tek işlemlerle uğraşmasına gerek kalmaz.



Şekil-2: Kullanıcı Arayüzü

Raporlama modülü, kapsamlı testlerin sonucunu belirli bir formatta kullanıcıya vermek için kullanılmaktadır. Sonuçlar daha anlaşılabilir bir düzenle, bir mantık

sırası içerisinde rapora yerleştirilmektedir. Belirlenen güvenlik açıkları işaretlenerek, kullanıcının dikkati çekilmektedir. Böylece kullanıcı dikkat etmesi gereken hususları daha kolay görebilmektedir. Uygun olan yerlerde açığı kapatmak için öneriler de sunulmaktadır. Sadece belirli testlerin çalıştırıldığı durumlarda raporlama modülü kullanılmamaktadır. Böyle testlerde yapılan testin sonucu doğrudan kullanıcı arayüzü vasıtasıyla kullanıcıya sunulmaktadır.

4. Karşılaştırma

Geliştirilen sistemin önemini anlamak açısından en iyi yöntem onu mevcut olan benzer araçlarla karşılaştırmaktır. Tablo 2'de, Ağ Güvenlik Açığı Tarayıcısının, mevcut sekiz adet ağ zafiyet tarayıcısı ile değişik açılardan karşılaştırması verilmiştir. Karşılaştırılanların bir kısmı ücretsiz araçlarken, bir kısmı ise ücretli araçlardır.

Tablo 2'nin satırlarında zafiyet tarama araçları tarafından desteklenen özellikler görülmektedir. Sütunlarda ise, ilk sütunda bu çalışma kapsamında geliştirilen sistem olmak üzere ücretli veya ücretsiz kullanılabilen dokuz araç görülmektedir. Karşılaştırılan araçlar tarafından desteklenen özellikler işaretlenmiş durumdadır. Tablonun son satırında araçların ücret durumları gösterilmiştir. Dolar işareti konulan araçlar ücretli araçlardır. Diğerleri ise internet üzerinden ücretsiz indirilip kullanılabilen araçlardır.

Tablo 2: Zafiyet tarayıcıların karşılaştırması

	AGAT	NMap	Nessus	GFI Lan Guard	Supper Scan	Solar Winds	F-Scan	Core Impact	MaxPatrol
Port tarama	√	√	√	√	√	√	√	√	√
Whois arama	√			√					√
Kaplosuz ağ taraması	√								
DNS&Subdomain taraması	√		√	√		√		√	√
Google mail araması	√								
Banner grabbing	√	√	√	√	√	√	√	√	√
Brute force attack	√		√	√		√		√	√
SSL taraması	√		√	√		√		√	√
Web zafiyet taraması	√		√					√	√
Web Ext. Taraması	√		√					√	√
URL bulucu	√							√	√
SNMP tarayıcı	√		√	√		√		√	√
SMTP kullanıcı tespiti	√		√	√		√		√	√
LAN haritası	√								
SMB tarayıcı	√		√	√		√		√	√
Web dayanıklık testi	√								
Ücret durumu				$	$	$		$	$

Desteklenen özellikler incelendiği takdirde, geliştirilen Zafiyet Tarama Sisteminin mevcut ürünlere göre daha fazla özellik içerdiği görülmektedir. Ücretli ürünler olan MaxPatrol, Core Impact gibi oldukça kapsamlı özelliklere sahip ürünler seviyesinde bir ürün ortaya çıkarılmıştır. Dahası onlarda olmayan bazı özellikler de ilave edilmiştir.

5. Sonuçlar

Günümüzde bilgisayar sistemleri çok daha güçlü ve kararlı sistemler haline gelmiştir. Fakat buna karşılık yönetimi de bir o kadar güçleşmiştir. Özellikle güvenlik açısından sistem ve ağların yönetimi her geçen gün daha da önemli bir konu haline gelmektedir. Kurumların, bir şekilde internete bağlı sistemlerini güvenli yapması gerekmektedir. Güvenlikten emin olmak için ise sistemlerin test edilmesi, var alan açıkların tespit edilmesi ve giderilmesi gerekir. Bu

işlemin de mümkün olduğunca ekonomik bir şekilde yapılması önemlidir.

Bilgisayar ve ağların güvenlik açıklarını veya zayıflıklarını ortaya çıkarmak üzere geliştirilmiş ücretsiz ve açık kaynak kodlu araçlar mevcuttur. Fakat bu araçlar belirli testleri yapmak üzere geliştirilmiş durumdalar ve çok kapsamlı testler için yetersiz kalmaktadırlar. Ayrı işlemler için ayrı araçların kullanımı ise gereksiz zaman kaybına sebep olmaktadır.

Bu çalışma kapsamında, ücretsiz ve açık kaynak kodlu güvenlik yazılımları bir araya getirilerek kapsamlı bir Ağ Güvenlik Açığı tarayıcısı geliştirilmiştir. Bazı taramalar için ise özel komut dosyaları oluşturulmuştur. Ayrıca, gerektiğinde işletim sistemi komutlarından da yararlanılmaktadır. Böylece ücretli ve ücretsiz bir çok araçtan daha fazla özelliğe sahip veya en az onlar kadar güçlü bir sistem geliştirilmiştir. Bir araya toplanan araçlar ortak bir grafiksel arayüz vasıtasıyla yönetilmektedir. Bu sayede, komut satırından verilen komutlarla çalıştırılan bazı araçlar kullanıcı dostu bir arayüze sahip olmuşlardır. Kullanıcı arayüzü vasıtasıyla belirli özellikler üzerinde tarama yapmak mümkün olduğu gibi tüm ağı kapsamlı bir kontrolden geçirmekte mümkündür. Sistemin çıkardığı raporlar, bir çok programın çıktısına göre daha anlaşılır bir formata sokulmuştur. Tespit edilen güvenlik açıkları işaretlenerek kullanıcının daha kolay görmesi sağlanmıştır. Ayrıca mümkün olan yerlerde yapılması gerekenlerle ilgili kullanıcıya ipuçları verilmiştir.

Son yıllarda ücretsiz ve açık kaynak kodlu yazılımları kullanma yönünde bir akım vardır. Gönüllü insanlar internet üzerinden bir araya gelerek oldukça profesyonel ürünler ortaya çıkarmaktadırlar. Güvenlikle ilgili olarak da ücretsiz ve açık kaynak kodlu geliştirilen araçlar mevcuttur. Ancak, bunlar tek tek ele alındıklarında zayıf kalmaktadırlar. Ayrı ayrı kullanılmaları gereksiz insan gücü ve zaman kaybına sebep olmaktadır. Bu çalışma kapsamında, ayrı ayrı yürütülen çalışmaların bir araya getirilmeleri halinde oldukça üstün özelliklere sahip ve ticari eşitlerine göre kıyaslanabilir ürünler ortaya çıkarılabileceği görülmektedir.

Ücretsiz ve açık kaynak kodlu yazılımları kullanarak geliştirilecek böyle bir sistemin en büyük sorunu sürdürülebilirliği olarak değerlendirilmektedir. Teknolojideki gelişmelere bağlı olarak kullanılan araçlar zaman zaman güncellenmektedir. Bunların takip edilerek sisteme yansıtılması ihtiyacı vardır. Diğer bir sorun ise bazı araçların geliştirme sürecine son verilmesidir. Bu durumlarda güncellemeler yapılamayacaktır. Oysa teknolojideki gelişmelerin takip edilerek araçların güncellenmesi oldukça önemlidir. Dolayısıyla ortaya çıkabilecek bu tür sorunlarda sistemi sürdürebilmek için, kullanılan araçlarda yapılan güncellemeleri yansıtabilecek, geliştirilmesi durdurulan ürünler yerine yenisini koyabilecek uzman veya uzmanların olmasına ihtiyaç vardır.

Bu çalışmanın devamında, yeni zafiyet tarama modüllerinin eklenmesi gelecek planlar arasında yer almaktadır. Güvenlik açıklarının tespiti için ayrıntılı sızma (penetration) testlerinin yapılması bir başka ileriye yönelik plandır. Geliştirilen sistemin bilgisayarlara kolayca kurulumunu sağlamak üzere bir kurulum programının oluşturulması da yapılacak faaliyetler arasında yer almaktadır.

6. Kaynakça

[1] S.Lipner, “The Trustworthy Computing Security Development Lifecycle”, Computer Security Applications Conference, (2004), pp. 2-13.

[2] Budiarto, R.; Sureswaran Ramadass; Samsudin, A.; Noor, S.; “Development of penetration testing model for increasing network security”, Proc. Of Int.Conf. on Information and Communication Technologies: From Theory to Applications, 19-23 April 2004.

[3] D.Davidson, “Thwarting Stealthy Attacks with Anti-Reconnaissance”, the ISSA Journal, June 2006.

[4] Debra S. Isaac and Michael J. Isaac, The SSCP Prep Guide: Mastering the Seven Key Areas of System Security, John Wiley & Sons, 2003.

[5] K.Scarfone, M.Souppaya, A.Cody, A.Orebaugh, Technical Guide to Information Security Testing and Assessment, National Institute of Standards and Technology, September 2008.

[6] http://sectools.org/vuln-scanners.html, Access date: 26 august 2009.

[7] M.S.Elliott, W.Scacchi, Free software developers as an occupational community: resolving conflicts and fostering collaboration, Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, Florida, USA, 2003.

[8] W.Scacchi, J.Feller, B.Fitzgerald, S.Hissam and K.Lakhani, Understanding Free/Open Source Software Development Processes, Softw. Process Improve. Pract. 2006; 11: 95–105.

[9] GNU Project, http://www.gnu.org/

[10] NMap, http://nmap.org/, Access date: 26 august 2009.

[11] AMap, http://freeworld.thc.org/thc-amap/, Erişim tarihi: 1 Kasım 2009.
[12] Dnsenum, http://code.google.com/p/dnsenum/, Access date: 26 august 2009.
[13] Nikto, http://www.cirt.net/nikto2, Erişim tarihi: 1 Kasım 2009.
[14] DIRB, http://dirb.sourceforge.net/, Access date: 26 august 2009.
[15] Samba Project, http://us1.samba.org/, Access date: 26 august 2009.
[16] BannerGrab, http://labs.portcullis.co.uk/application/ bannergrab/, Access date: 26 august 2009.
[17] Subdomainer, http://www.edge-security.com/ subdomainer.php, Access date: 26 august 2009.
[18] OneSixtyOne, http://www.phreedom.org/software/onesixtyone/, Access date: 26 august 2009.
[19] Curl-loader, http://curl-loader.sourceforge.net/ #overview, Access date: 26 august 2009.
[20] KISMET, http://www.kismetwireless.net/, Access date: 26 august 2009.
[21] Lanmap, http://parseerror.com/lanmap/, Access date: 26 august 2009.
[22] Kommander, http://kommander.kdewebdev.org/, Access date: 26 august 2009.