

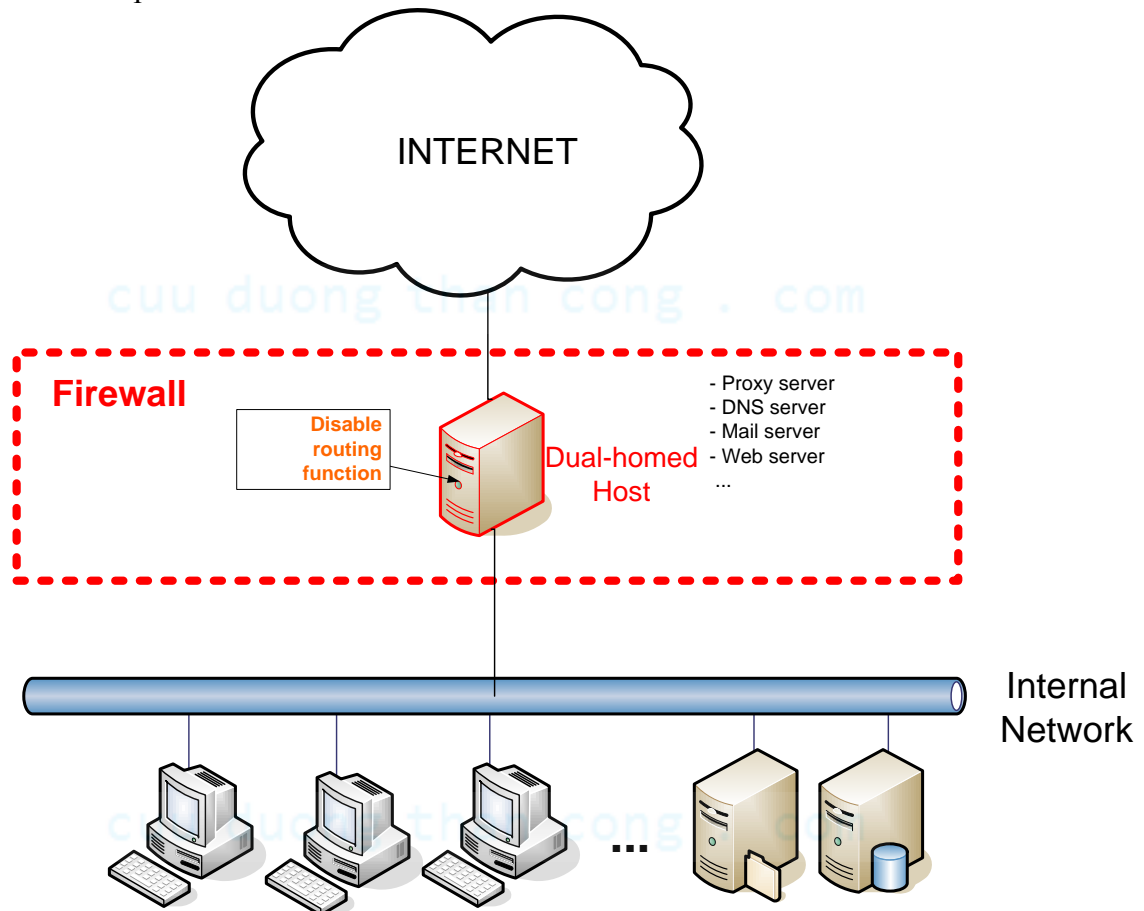
MỘT SỐ GỢI Ý TRONG VIỆC BẢO MẬT VẬT LÝ - MẠNG – HỆ ĐIỀU HÀNH – CSDL - ỨNG DỤNG

1. MẠNG :

1.1. Các kiến trúc firewall căn bản :

1.1.1. Dual-Homed Host

- Phải disable chức năng routing của dual-homed host để cấm hoàn toàn lưu thông IP từ ngoài vào.
- Các hệ thống bên trong và bên ngoài dual-homed host chỉ có thể liên lạc với dual-homed host mà chúng không liên lạc trực tiếp được với nhau.
- Dual-homed host cung cấp dịch vụ thông qua proxy server hoặc login trực tiếp vào dual-homed host.

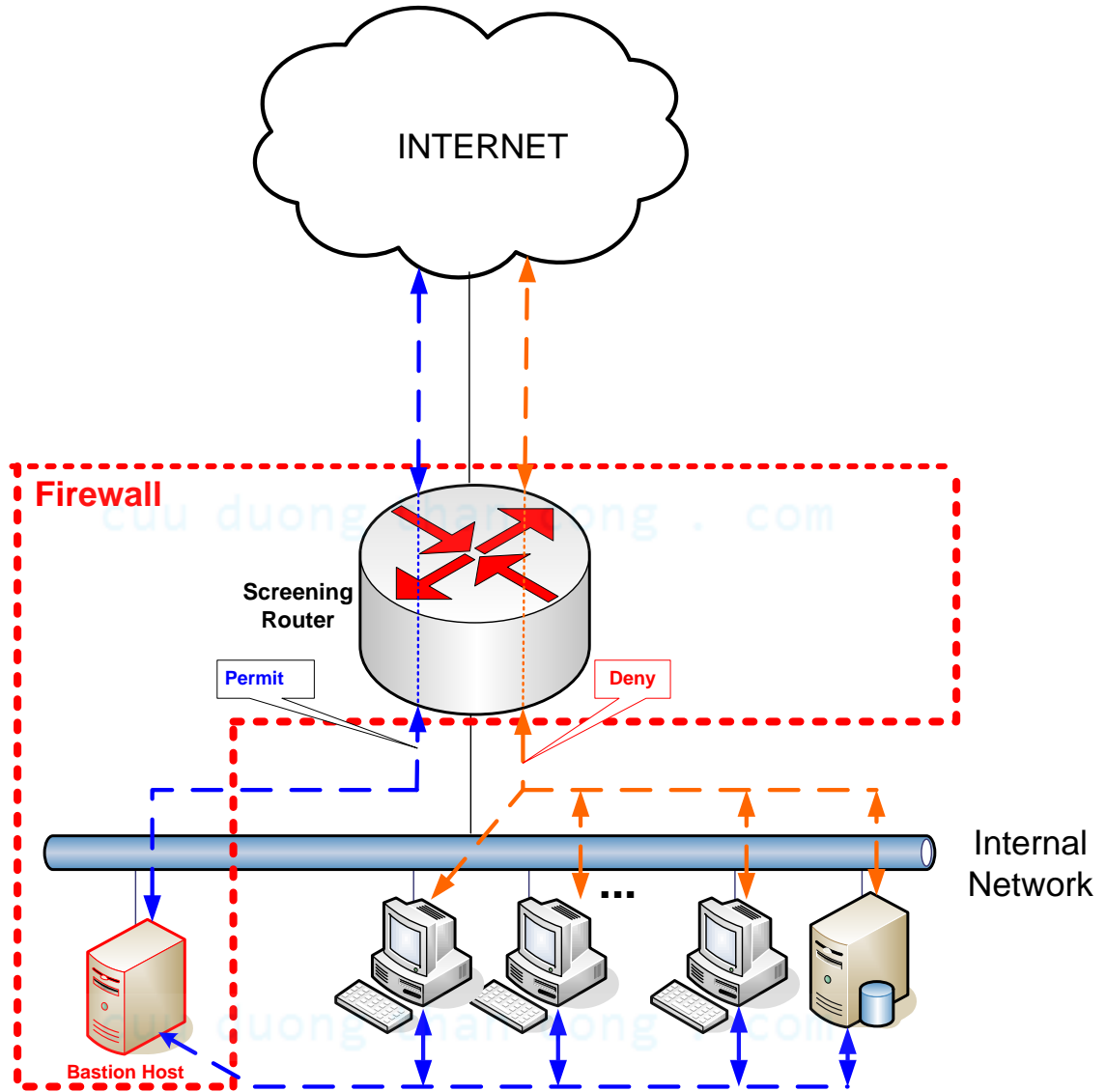


1.1.2. Screened Host

- Trong kiến trúc này chức năng bảo mật chính được cung cấp bởi chức năng packet filtering tại screening router.
- Packet filtering trên screening router được setup sao cho bastion host là máy duy nhất trong internal network mà các host trên internet có thể mở

kết nối đến. Packet filtering cũng cho phép bastion host mở các kết nối (hợp pháp) ra bên ngoài (external network).

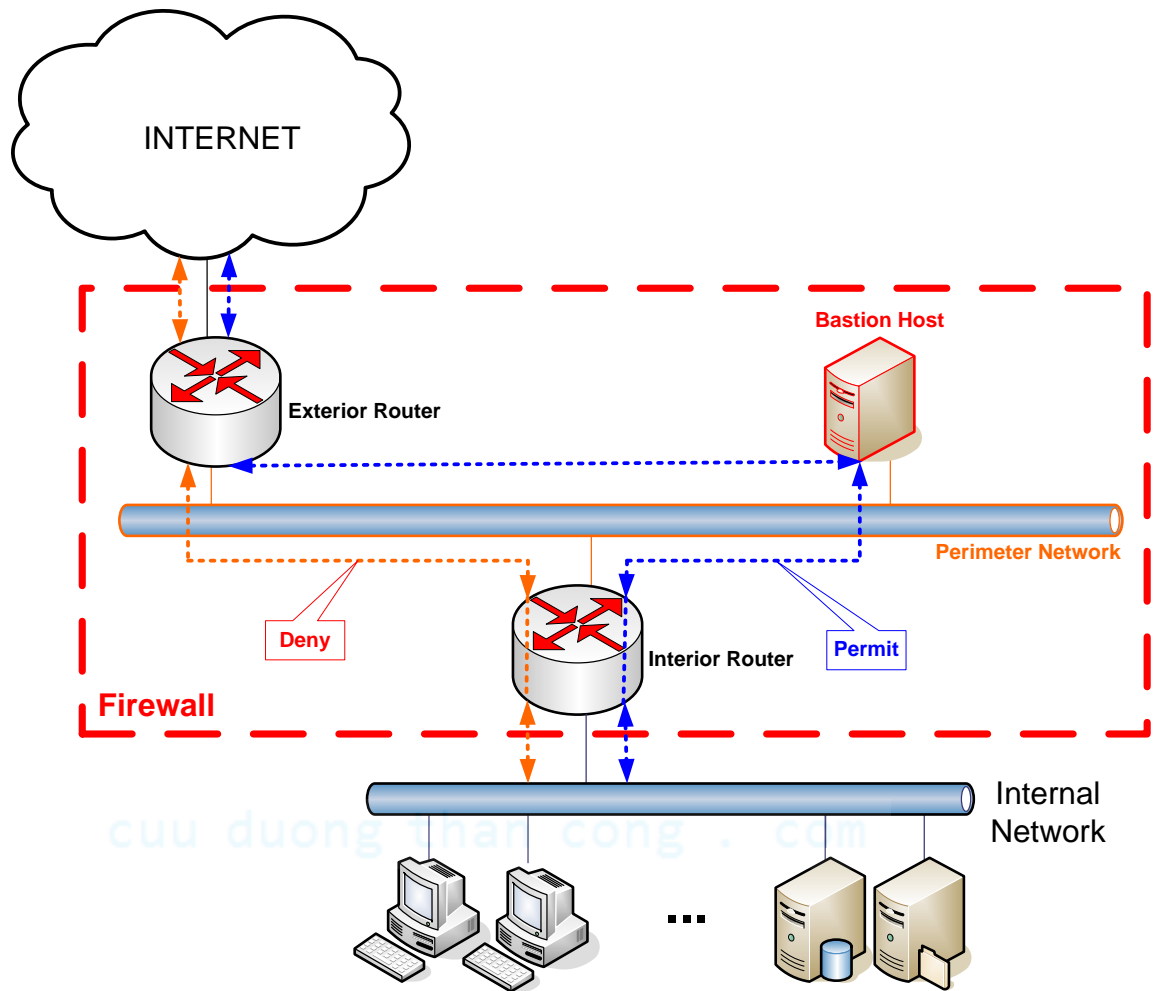
- Thường Packet filtering thực hiện các công việc như sau :
[1]. Cho phép các internal hosts mở kết nối đến các host trên internet đối với 1 số dịch vụ được phép.
[2]. Cấm tất cả kết nối từ các internal hosts
- Khi hacker đã tấn công được vào bastion host thì không còn một rào chắn nào cho các internal hosts.



1.1.3. Screened Subnet

- Thêm 1 perimeter network để cô lập internal network với internet. Như vậy dù hacker đã tấn công được vào bastion host vẫn còn 1 rào chắn nữa phải vượt qua là interior router. Các lưu thông trong internal network được bảo vệ an toàn cho dù bastion đã bị “chiếm”.
- Các dịch vụ nào ít tin cậy và có khả năng dễ bị tấn công thì nên để ở perimeter network.
- Bastion host là điểm liên lạc cho các kết nối từ ngoài vào như : SMTP; FTP; DNS. Còn đối với việc truy cập các dịch vụ từ internal clients đến các server trên internet thì được điều khiển như sau :
 - + Set up packet filtering trên cả hai exterior và interior router để cho phép internal clients truy cập các servers bên ngoài 1 cách trực tiếp.
 - + Set up proxy server trên bastion host để cho phép internal clients truy cập các servers bên ngoài 1 cách gián tiếp.
- Nên hạn chế các dịch vụ mà interior router cho phép giữa bastion host và internal net để giảm đi số máy có nguy cơ bị tấn công tiếp theo khi bastion đã bị “chiếm”.
- Exterior router cho phép tất cả lưu thông từ perimeter net ra internet. Các packet filtering rules thiết yếu để bảo vệ cho các internal hosts là giống nhau trên tại exterior router và interior router. Thường exterior router thực hiện packet filtering rules tổng quát, chung chung, ít chi tiết hơn so với interior router (ngoại trừ những packet filtering rules thật thiết yếu thì giống nhau). Việc phát hiện và ngăn cấm sự giả mạo địa chỉ được thực hiện tại exterior router.

cuu duong than cong . com

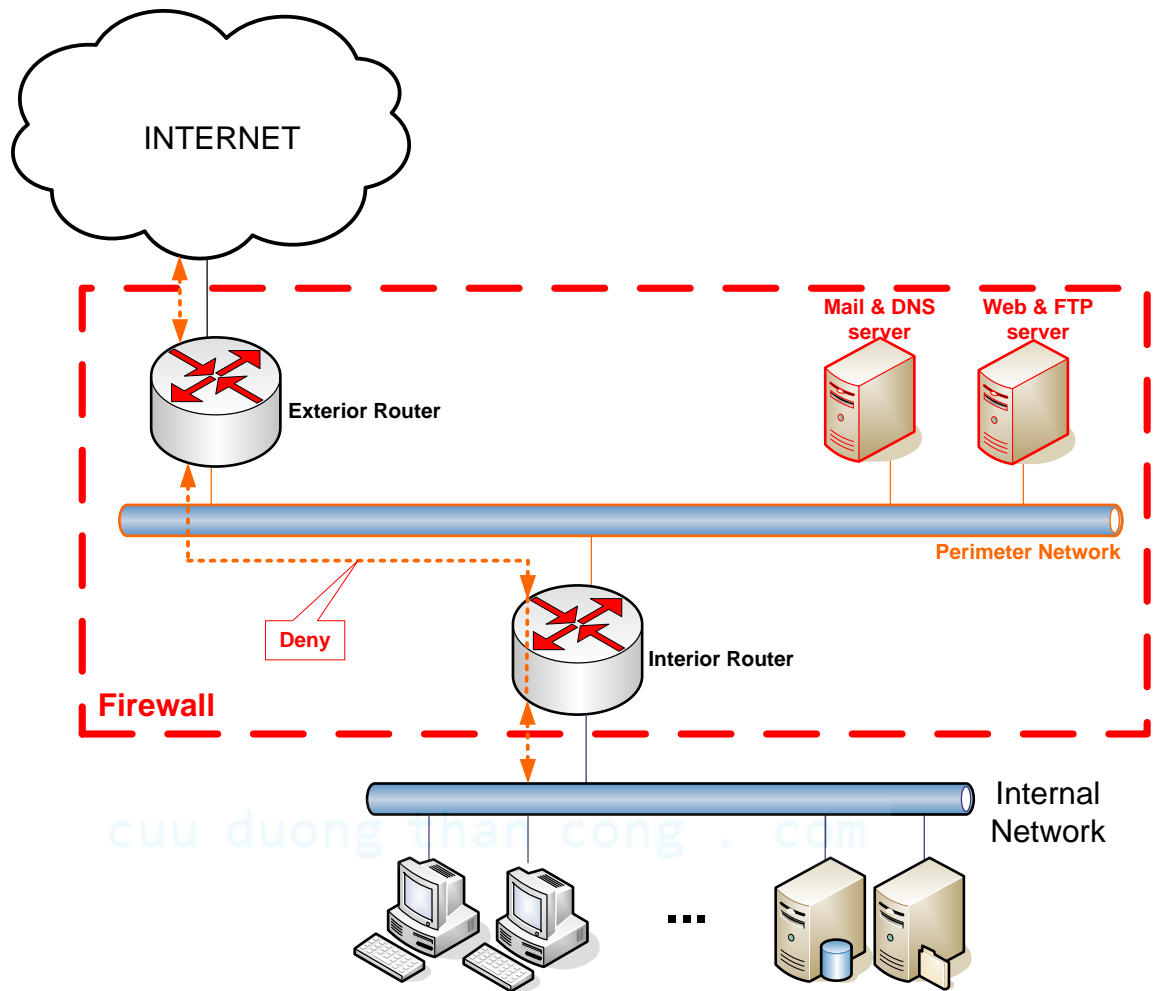


1.2. Một số lưu ý về các kiến trúc firewall :

1.2.1. Các dạng kiến trúc firewall khác có thể dùng

1. Dùng nhiều Bastion Hosts

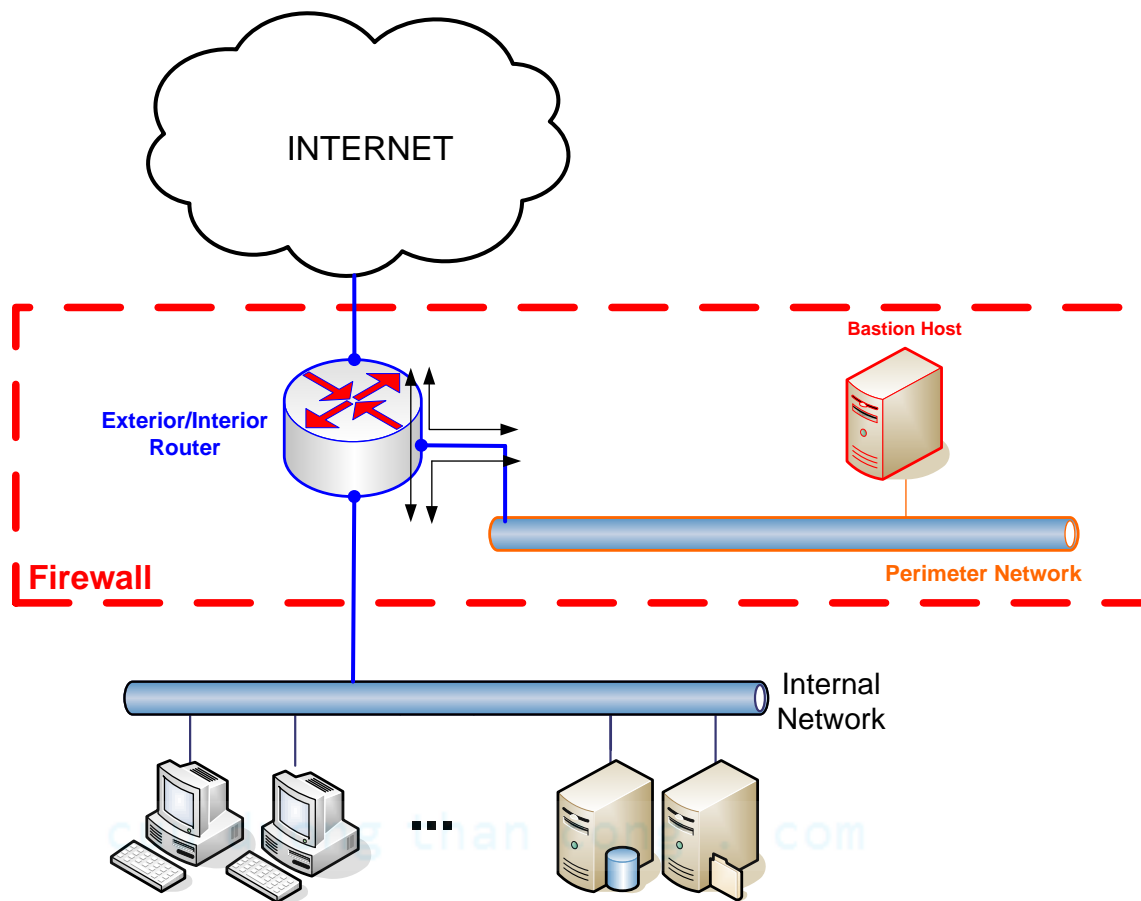
Để tăng performance, redundancy và tách biệt các servers và dữ liệu



2. Ghép Interior Router với Exterior Router

- Router phải cho phép áp dụng các luật cho dòng packet đi vô và đi ra trên mỗi interface.

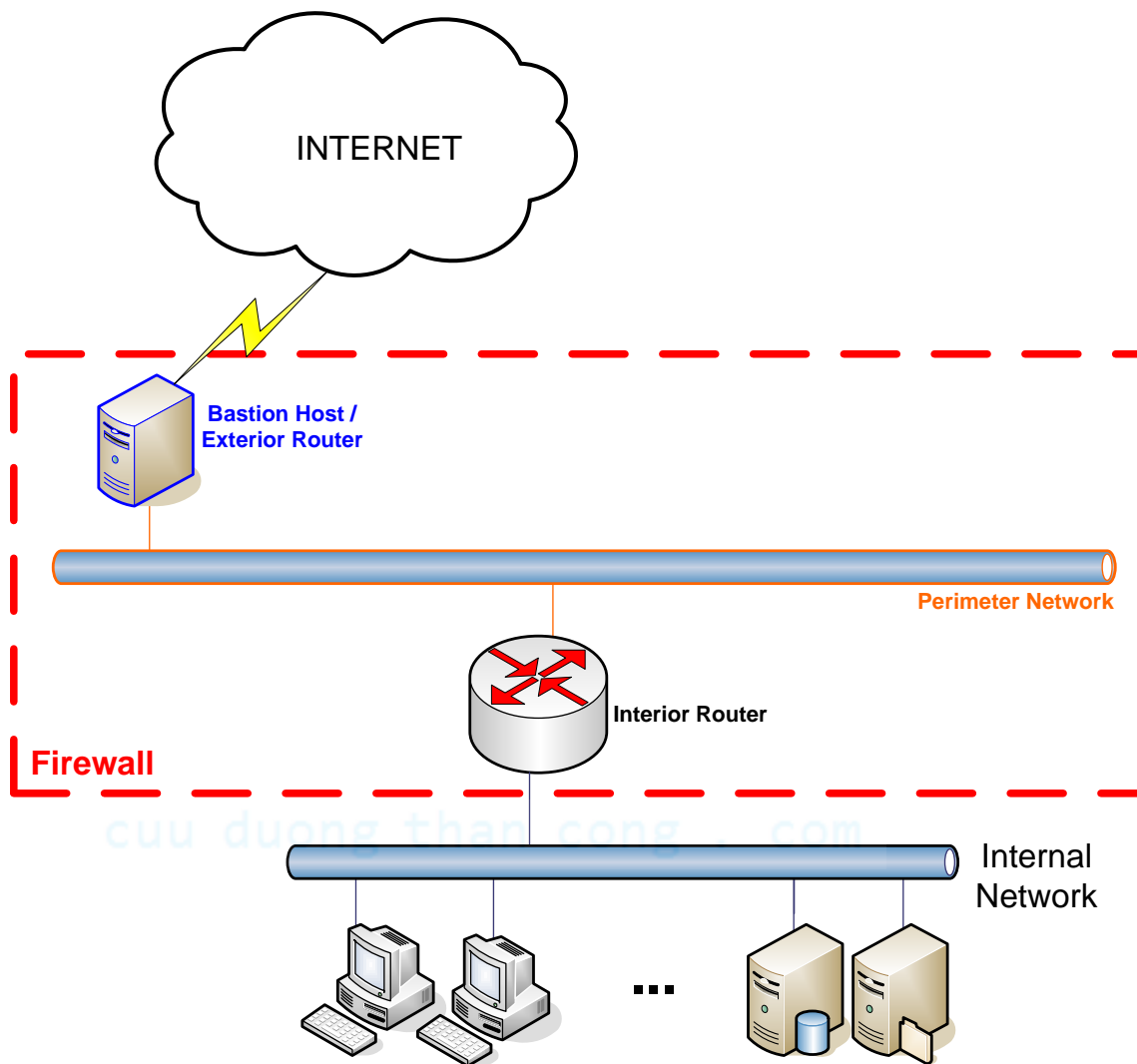
cuu duong than cong . com



3. Ghép Bastion Host và Exterior Router

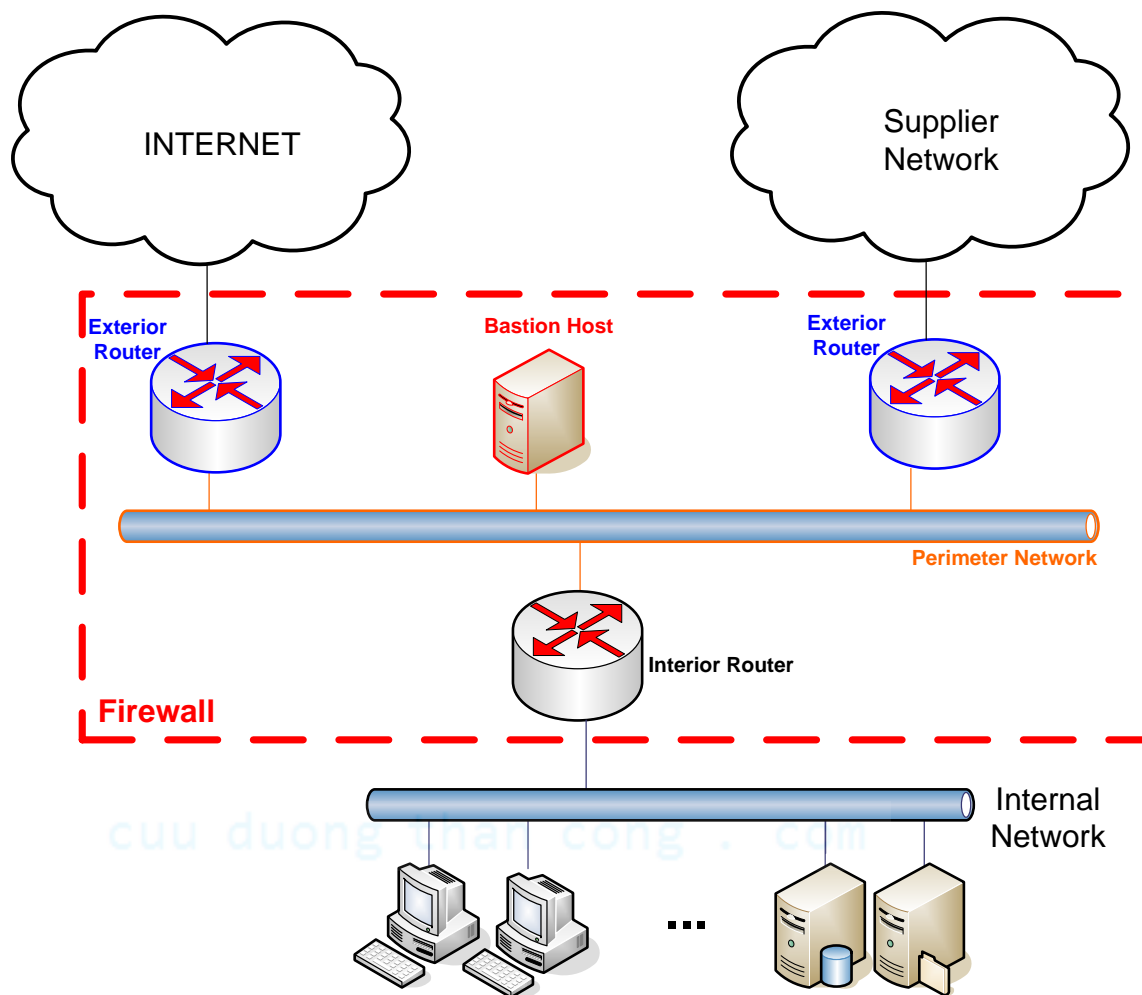
Thường được dùng trong trường hợp dùng kết nối PPP lên internet

cuu duong than cong . com



4. Dùng nhiều Exterior Routers

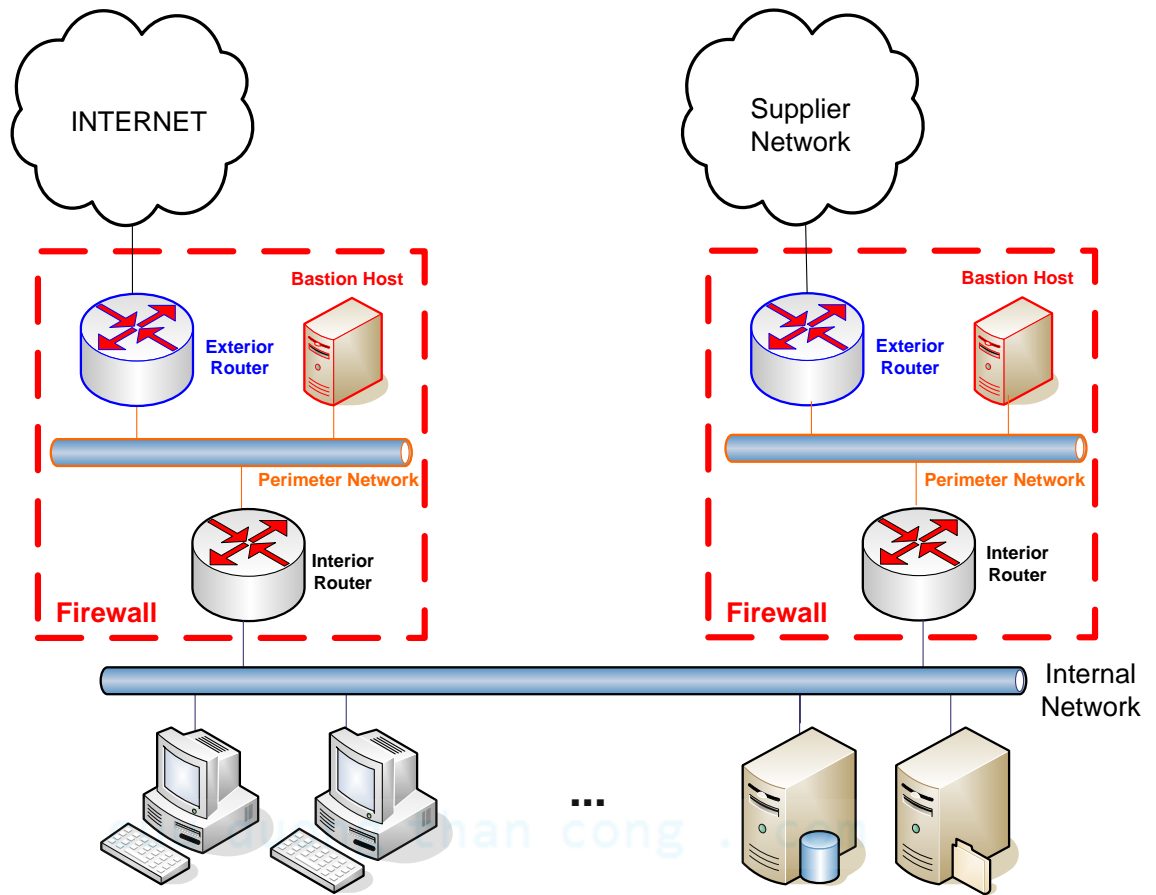
Trong trường hợp có nhiều kết nối lên internet hoặc trường hợp 1 kết nối lên internet và các kết nối đến các mạng bên ngoài khác.



5. Dùng nhiều Perimeter Networks

Dùng nhiều perimeter net để cung cấp đặc tính dư thừa (redundancy) cho hệ thống.

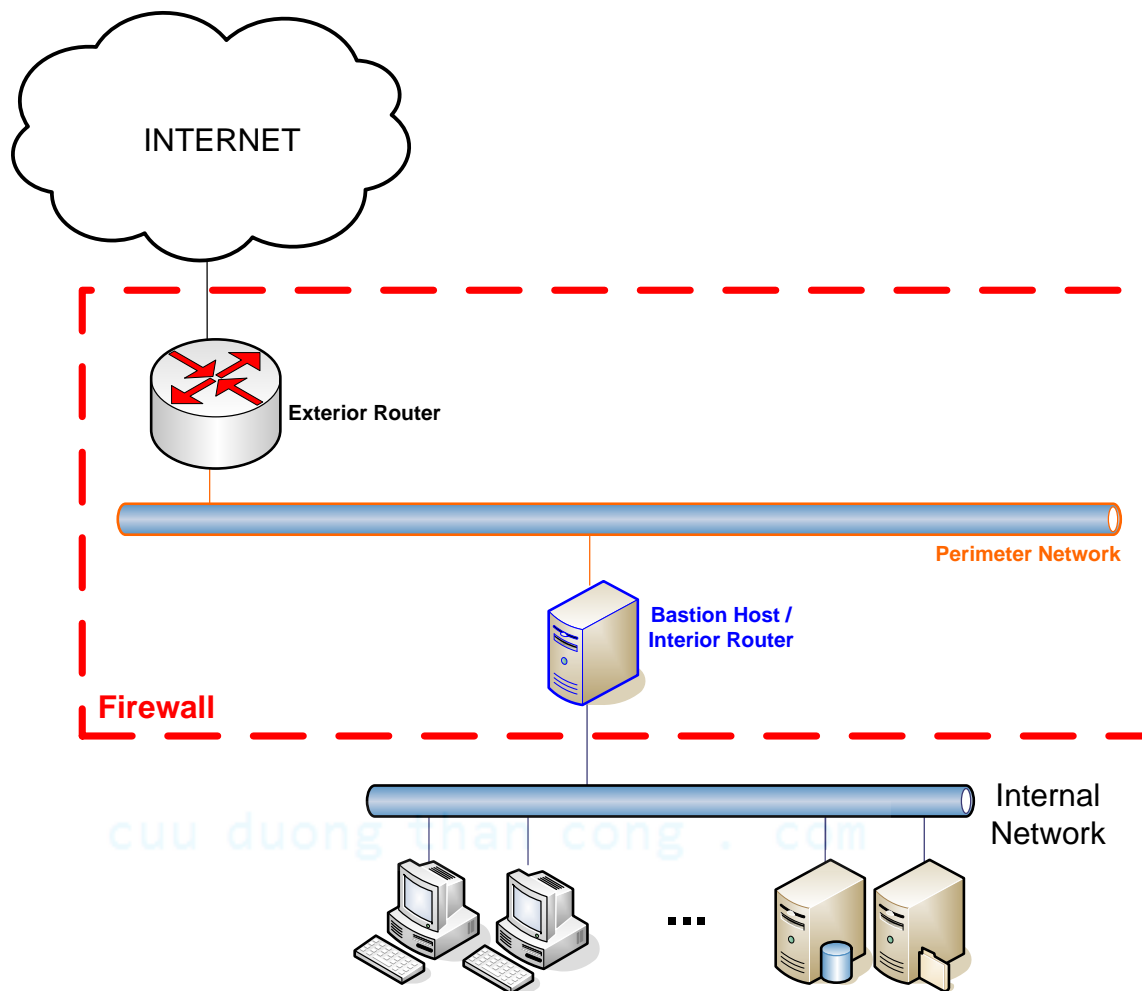
cuu duong than cong . com



1.2.2. Các kiến trúc không nên dùng

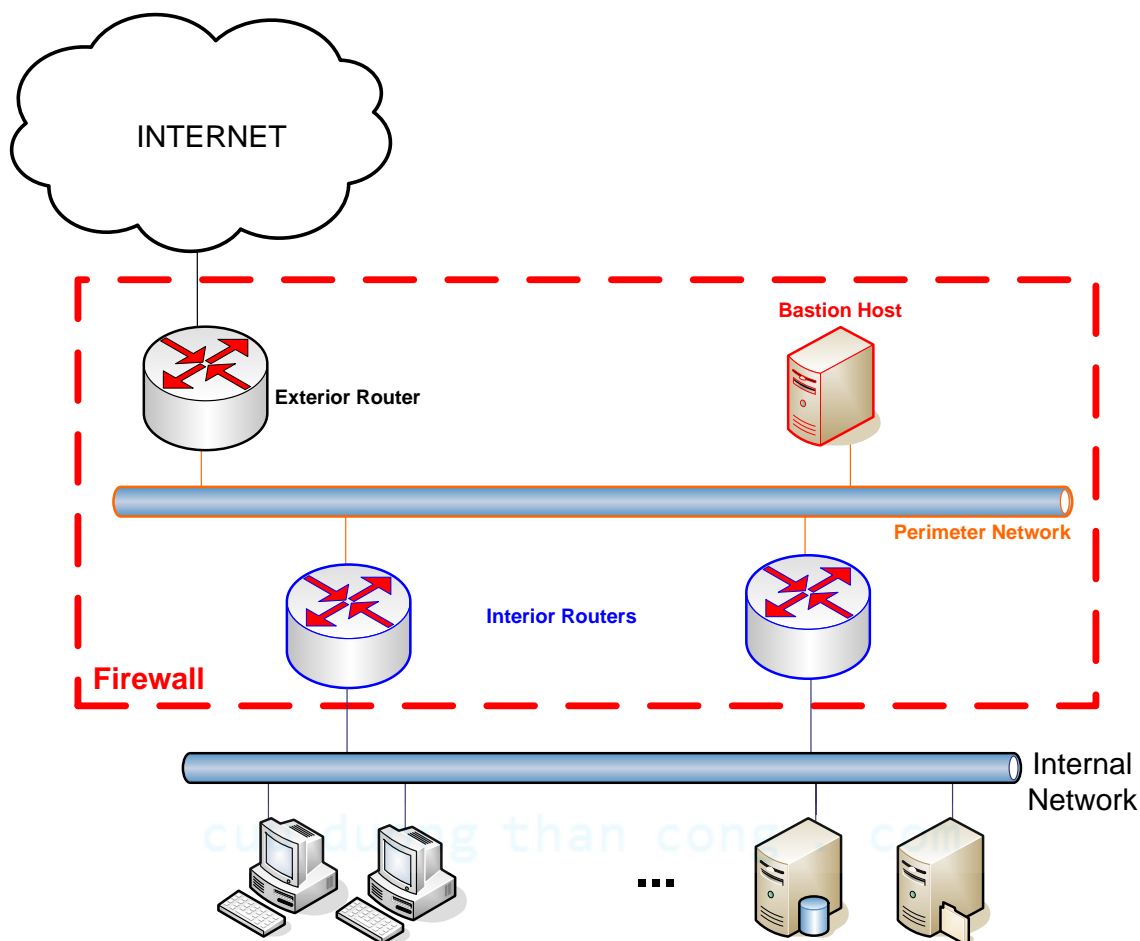
1. Ghép Bastion Host và Interior Router

cuu duong than cong . com



2. Dùng nhiều Interior Routers

cuu duong than cong . com



1.3. Một số lưu ý đối với máy giữ vai trò Bastion Host :

1. Cấm các user accounts
2. Tắt bỏ các dịch vụ không cần thiết
3. Cài đặt phiên bản hệ điều hành “gọn gàng & sạch sẽ” nhất có thể được
4. Sửa tất cả các lỗi hệ thống
5. System logs
6. Tắt bỏ chức năng routing

2. HỆ ĐIỀU HÀNH :

Những lỗi phổ biến thường gặp khi sử dụng HĐH là :

- **HĐH có lỗi :** Không quan tâm hay không kịp khắc phục các lỗ hổng của HĐH sẽ dẫn đến việc dễ dàng bị xâm nhập.
- **Cấu hình sai HĐH :** Do sơ sót hay thiếu kiến thức đưa đến việc cấu hình HĐH sai, cho phép vận hành các dịch vụ nên cấm.
- **Virus**

Gợi ý khắc phục :

- **Chứng thực tập trung :** SSO(single sign on) là giải pháp quản lý chứng thực tập trung tại 1 nơi giúp giảm đi sự nhầm lẫn. Ví dụ : LDAP ...

- Cập nhật thường xuyên các bản vá lỗi
- Cấm tất cả các dịch vụ không cần thiết
- Ghi nhật ký các hành động xảy ra (log file) trên hệ thống
- Thường xuyên cập nhật phần mềm Anti-virus
- Sử dụng hiệu quả các công cụ bảo vệ có sẵn
- Kiểm tra hệ thống liên tục (ví dụ Systems Scanner)

3. ỨNG DỤNG WEB :

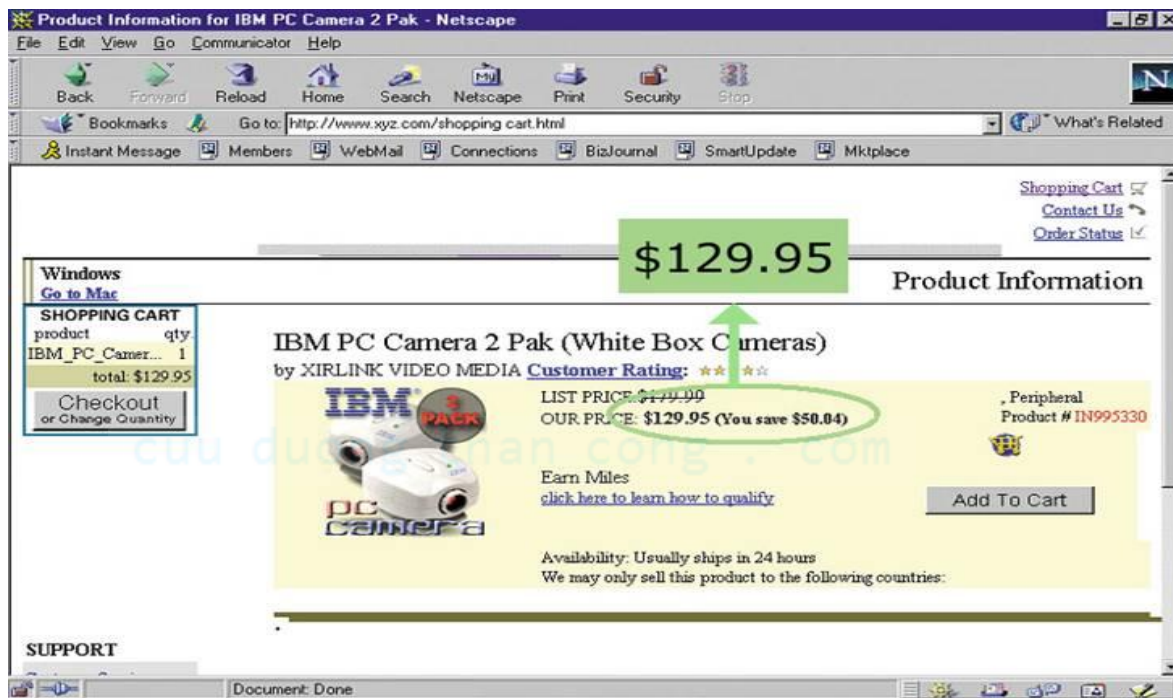
Sơ lược về những cách tấn công phổ biến hiện nay :

1. Hidden Manipulation - Thao tác vùng ẩn

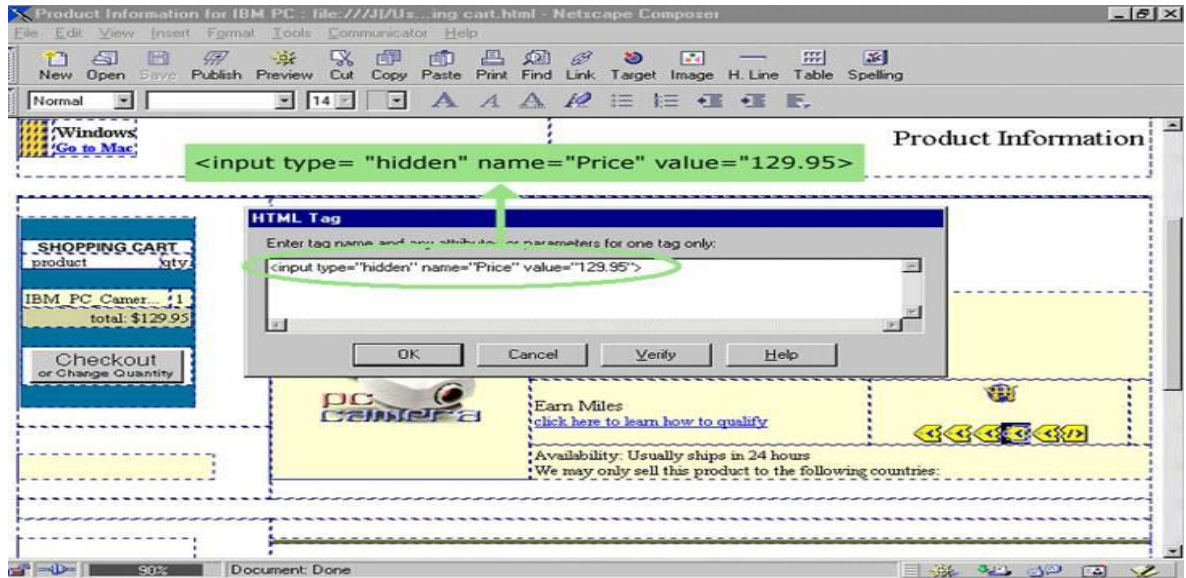
Các phân bị giấu đi trong trang Web thường được dùng để lưu thông tin về phiên làm việc của client, phiên làm việc này được ghi nhớ ở máy client chứ không cần phải tổ chức CSDL phức tạp trên server. Tuy vậy, phân bị giấu đi này không "ẩn" thực sự, chức năng "View Source" của trình duyệt cho phép đọc được mã nguồn của phân bị giấu của trang Web. Dựa vào mã nguồn này tin tặc có thể giả lập phiên làm việc để truy cập thông tin trên máy chủ hay tìm ra sơ hở của trang WEB mà ta muốn tấn công,...

Ví dụ :

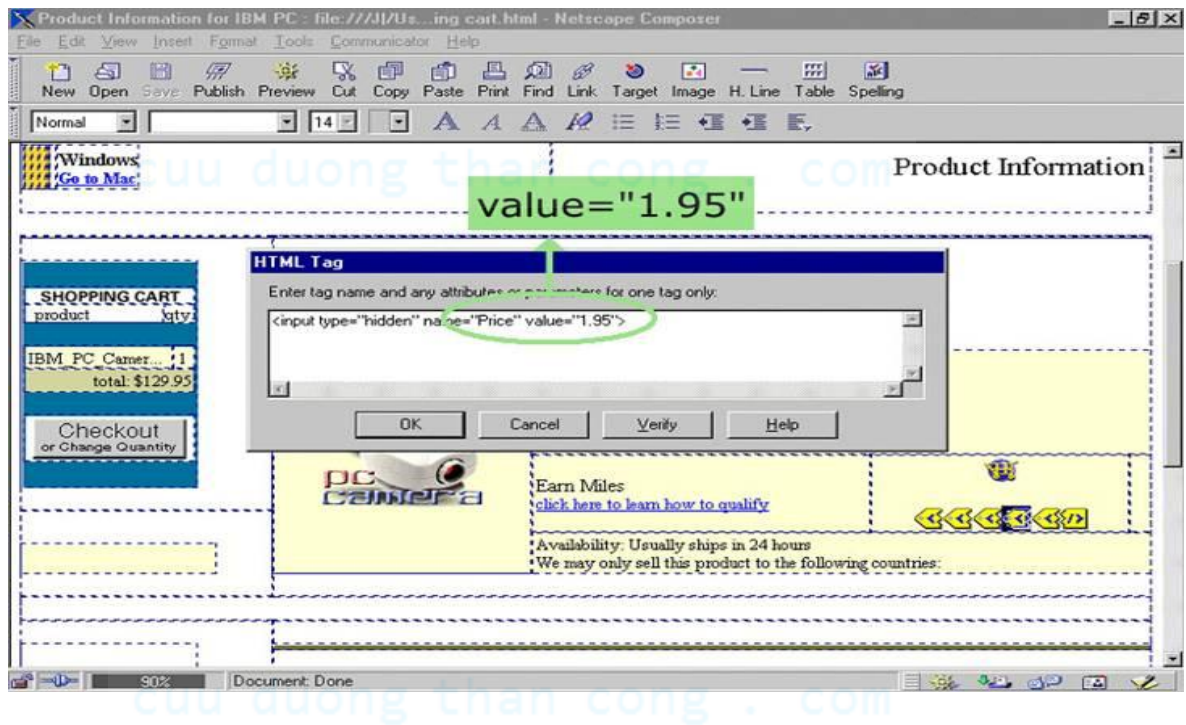
[1].



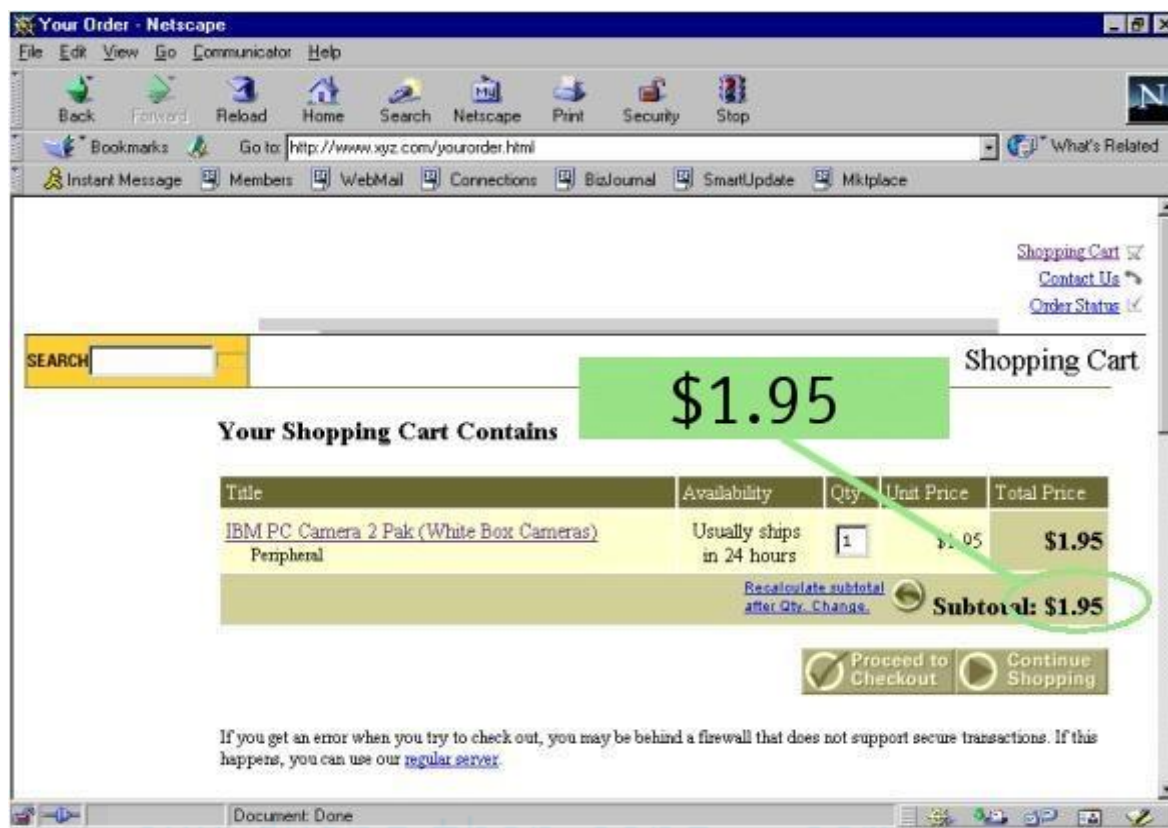
[2].



[3].



[4].



Loại tấn công này có thể phòng chống bằng cách:

Phải chắc chắn rằng form được nhập phải từ trang gọi đoạn script. Phải kiểm tra đoạn script không được thực thi trực tiếp bởi URL. Tuy nhiên, chúng ta không thể tin cậy trọn vẹn bởi 2 lý do: Khi HTTP_REFERERER là một thuộc tính của trình duyệt, điều này dễ dàng bị thay đổi bởi tin tặc. Hơn thế nữa, một vài người dùng là người vô danh (anonymous) thì kết quả không có giá trị cho HTTP_REFERERER. Đối với anonymous ta dành một dịch vụ riêng cho phép người dùng xem các trang web mà không cho bất kì ai có thu thập thông tin về các site này và những site này cũng không có quyền lấy thông tin của họ, chẳng hạn như là địa chỉ IP. Tăng cường xử lý và hiệu quả các trường nhập giá trị của người dùng, chờ thông tin nhập vào (ví dụ như: số, chữ), những đặc tính khác và những gì có liên quan đến người dùng.

2. Buffer Overflow Attacks - Tràn bộ đệm

Tình trạng tràn bộ đệm xảy ra khi dữ liệu được gửi đến ứng dụng nhiều hơn mong đợi. Kỹ thuật tấn công này có thể làm cho hệ thống bị tê liệt hoặc làm cho hệ thống mất khả năng kiểm soát.

Ví dụ: một chương trình cần nhập dữ liệu của người dùng với kích thước tối đa 256 bytes. Hệ điều hành sẽ cấp phát một vùng trong bộ nhớ máy tính (đó chính là một bộ đệm - buffer) dành cho chương trình để lưu trữ 256 bytes hoặc ít hơn. Nếu người dùng nhập vào quá 256 bytes và chương trình không kiểm tra điều này, tràn

bộ đệm sẽ xảy ra. Vì chương trình máy tính cần không gian để lưu trữ những byte dư ra, nó sẽ chứa lên những vùng nhớ kế cạnh và ghi đè lên những dữ liệu có sẵn tại đó.

Còn có một bộ đệm khác trên bộ nhớ máy tính dùng để lưu trữ địa chỉ cho lệnh máy kế tiếp sẽ được thực thi sau khi gọi hàm. Vùng nhớ này được cấp phát trên stack và được gọi là con trỏ lệnh (instruction pointer). Tiếp tục ví dụ trên, giả sử sau khi đọc vào dữ liệu nhập của người dùng, chương trình sẽ thực hiện lệnh in ra nội dung. Con trỏ lệnh khi đó sẽ có giá trị là địa chỉ vùng nhớ của lệnh in. Máy tính sẽ thực hiện các thao tác tuần tự như sau: đọc dữ liệu nhập của người dùng, lưu trữ nó vào bộ đệm, kiểm tra con trỏ lệnh để biết lệnh thực thi kế tiếp, tìm địa chỉ vùng nhớ của lệnh in, đọc nội dung của bộ đệm và in nó ra màn hình.

Bây giờ hãy kết hợp lại, nếu tin tặc có thể làm tràn bộ đệm sao cho thay đổi nội dung của con trỏ lệnh bằng cách trỏ đến đoạn mã lệnh của mình, anh ta có thể làm được nhiều chuyện khác. Và đó là những gì diễn ra trong thực tế. Tin tặc làm tràn bộ đệm sao cho con trỏ lệnh sẽ trỏ đến một đoạn mã tạo ra một giao tiếp dòng lệnh (command line, ví dụ /bin/sh). Sau khi chương trình thực hiện làm tràn bộ đệm, nó sẽ tìm đến địa chỉ của đoạn mã trên để thực thi tiếp. Nếu chương trình được chạy dưới quyền của người quản trị, tin tặc đã có được một giao tiếp dòng lệnh với quyền tương đương và có thể điều khiển toàn bộ hệ thống.

Dưới đây là một số đề xuất, hướng dẫn và tiện ích có thể sử dụng để phòng chống :

- *Luôn sử dụng kiểm tra giới hạn khi viết chương trình*
- *Xem xét lại tính bảo mật của mã nguồn các phần mềm kế thừa*
- *Tránh sử dụng các hàm không cung cấp kiểm tra giới hạn trong ngôn ngữ C, thay vào đó là các hàm tương đương. Thay các hàm gets, strcpy, strcat, sprintf, scanf, sscanf bằng các hàm tương đương fgets, strncpy, strncat, bcopy, bzero, memcpy.*
- *Sử dụng các vùng nhớ được cấp phát động.*
- *Cẩn thận khi sử dụng các vòng lặp để sao chép dữ liệu từ giữa các biến, cần đảm bảo giới hạn đã được kiểm tra.*
- *Sử dụng các tiện ích như StackGuard, StackShield để bảo vệ vùng bộ nhớ stack khỏi tràn bộ đệm.*
- *Sử dụng các công cụ và hướng dẫn để đánh giá mức độ an toàn của chương trình như Slint, rats, its, flawfinder.*
- *Cài đặt ngay các bản sửa lỗi.*

3. Parameter Tampering - Chèn tham số

Đây là cách thức tấn công bằng cách đưa tham số trực tiếp vào địa chỉ URL để truy cập thông tin không dành cho người dùng (người dùng thao tác qua giao diện trên trình duyệt không thể thấy được các thông tin này). Câu lệnh SQL truy cập CSDL nền của ứng dụng trên mạng thường được thể hiện trên địa chỉ URL. Tin tặc có thể thao tác trên đoạn mã SQL này để truy cập thông tin trong CSDL. Thường thao tác tham số có thể thực hiện với:

[1]. HTML Form Field Manipulation :

Khi người dùng thao tác trên trang web, thì thông tin được chọn đó sẽ lưu vào giá trị của biểu mẫu, và sẽ được gửi về ứng dụng như một HTTP request (GET hay POST). HTML cũng có thể lưu giá trị trong Hidden Field, những giá trị được lưu trong Hidden Field sẽ không được hiển thị trên màn hình.

Người dùng có thể thay đổi thuộc tính tất cả các loại biểu mẫu để nhập bất cứ giá trị nào họ muốn. Chẳng hạn như họ chỉ cần lưu trang web đó lại, chọn “view source” để xem và sửa nội dung sau đó chọn “save” và chạy trang web đó lại trên trình duyệt.

Ví dụ: ứng dụng dùng một biểu mẫu để nhập vào tên người dùng và mật khẩu sau đó gửi tới một CGI để xác thực bằng HTTP trên SSL. Một số người phát triển ứng dụng sẽ giới hạn chiều dài của tên người dùng và mật khẩu nhập vào bằng cách thiết lập giá trị “maxlength” để ngăn chặn bị đầy buffer do tin tặc có thể nhập vào một chuỗi rất dài. Tuy nhiên với cách phòng ngừa như trên thì tin tặc chỉ cần lưu trang này web lại, bỏ đi giới hạn chiều dài và chạy lại trên trình duyệt.

Còn Hidden Form Filed cho thấy sự tiện lợi khi dùng để lưu dữ liệu trên trình duyệt và là một trong những cách thông dụng nhất để lưu trữ dữ liệu từ trang này qua trang khác trong cùng một ứng dụng.

Ví dụ cùng ứng dụng có form để nhập tên người dùng và mật khẩu như trên thì sau khi đăng nhập có thể sẽ có một thẻ HTML như sau:

```
<input name="quyen_quan_tri" type="hidden" value="N">
```

Với tình huống này tin tặc có thể thay đổi giá trị value thành “Y”, thì ứng dụng sẽ xem như người dùng mới đăng nhập này là người quản trị ứng dụng. Hidden form field còn được dùng cho nhiều mục đích khác nữa, nên sẽ vẫn có chỗ hở mà tin tặc có thể lợi dụng.

Kĩ thuật phòng chống

Người thiết kế ứng dụng có thể dùng một biến session để tham chiếu đến thông tin được lưu trữ trên cache của server. Khi ứng dụng cần kiểm tra thông tin người dùng, ứng dụng sẽ so sánh giá trị session với giá trị trong bảng session trên server và sẽ chỉ đến thông tin của người dùng đó trong cache hay cơ sở dữ liệu.

Nếu không thể thực hiện theo cách trên ta có thể thực hiện cách sau:

- Ghép tên và giá trị của form hidden field thành một chuỗi đơn. Khi đó khoá sẽ được giấu kĩ. Chuỗi này có thể gọi là Outgoing Form Message. Sử dụng thuật toán mã hoá MD5 hoặc một kiểu hash một chiều khác để tổng hợp chuỗi Outgoing Form Message. Gọi là Outgoing Form Digest và lưu nó vào một hidden field.
- Khi giá trị trong biểu mẫu được submit, các thao tác như trên được thực hiện lại với cùng một khoá mà ta định trước ta có chuỗi Incoming Form Message rồi mã hoá thành Incoming Form Digest. Sau đó đem so sánh với Outgoing Form Digest, nếu chúng không khớp nhau thì chứng tỏ giá trị trong biểu mẫu đã bị thay đổi.

[2]. URL Manipulation :

Kiểu tấn công URL Manipulation cũng gần giống như kiểu lợi dụng Hidden Form Field để tấn công. Khi nhập một form HTML thì kết quả sẽ được gửi đi theo hai cách : GET hay POST. Nếu dùng GET, thì tất cả các tên biến và giá trị của nó sẽ xuất hiện trong chuỗi URL.

Ví dụ: có một trang web cho phép thành viên đã được thay đổi mật khẩu. Với trường hợp bình thường, thì thay đổi của người dùng sẽ được ghi nhận khi ấn nút submit. Và mệnh lệnh được gửi theo HTTP request

```
http://www.nganhang.com/example?user=thang&newpass=123fgf
```

Nhưng với một tin tặc có thể lợi dụng điều này để thay đổi mật khẩu của admin bằng cách thay đổi tham số như sau:

```
http://www.nganhang.com/example?user=admin&newpass=111111
```

Như thế những tham số mới này sẽ được gửi về ứng dụng để xử lý.

Không chỉ có các trang web HTML mới bị tấn công kiểu này. Hầu hết tất cả liên lạc trên internet đều dùng hyper link. Khi người dùng nhấp chuột lên một hyperlink để chuyển sang một trang khác hoặc ngay trong trang đó thì có nghĩa anh ta đã gửi một yêu cầu GET. Rất nhiều yêu cầu sẽ có chuỗi truy vấn với các tham số như một biểu mẫu. Do đó tin tặc có thể xem và thay đổi chúng.

Kĩ thuật phòng chống

Khi cần gửi tham số từ máy khách lên máy chủ, thì nên kết hợp với một session token để kiểm tra. Session token có thể là một tham số hoặc một cookie và Session token cũng đã có chế độ bảo mật riêng của nó.

Trong ví dụ trên, trước khi xử lý việc thay đổi mật khẩu thì Session token sẽ được kiểm tra xem người gửi yêu cầu có trong phiên làm việc của người đang thay đổi của mật khẩu hay không. Điều đó có nghĩa là người gửi yêu cầu có phải là admin hay không, nếu không phải thì không được phép thay đổi mật khẩu.

Giải pháp tốt nhất là tránh sử dụng tham số trong chuỗi truy vấn (đối với cả hidden field). Với nhiều tham số không nên cho người dùng thấy giá trị ví dụ là mật khẩu vì có thể có một người khác đang đứng sau người dùng cũng sẽ thấy được mật khẩu và ngoài ra trình duyệt cũng thường lưu lại các địa chỉ này.

Nếu không dùng hai cách trên thì ta có thể dùng cách mã hoá các tham số, cách này có ưu điểm bảo mật hơn nhưng cách cài đặt và xử lý phức tạp hơn hai cách trên rất nhiều.

[3]. HTTP Header Manipulation :

HTTP Header là thông tin điều khiển từ các yêu cầu HTTP của web client đến web server, và các phản hồi từ web server đến web client. Mỗi header thông thường bao gồm một dòng đơn của ASCII text với tên và dữ liệu. Sau đây là 1 ví dụ về header:

Host: www.someplace.org

Pragma: no-cache

Cache-Control: no-cache

User-Agent: Lynx/2.8.4dev.9 libwww-FM/2.14

Referer: http://www.someplace.org/login.php

Content-type: application/x-www-form-urlencoded

Content-length: 49

Thông thường các HTTP header được sử dụng bởi các trình duyệt và các phần mềm web server. Hầu hết các ứng dụng web thì không quan tâm đến nó. Tuy nhiên vẫn có một vài nhà thiết kế web dùng HTTP header trong ứng dụng mạng của mình, điều đó có thể vô tình tạo ra một lỗ hổng trong trang web. Nhờ đó tin tặc có thể lợi dụng để phá hoại bằng cách chỉnh sửa nội dung của các HTTP header. Cho dù những trình duyệt thông dụng không cho phép thay đổi các header. Tin tặc có thể tự viết một chương trình (khoảng 15 dòng Perl) để xem HTTP request, hay sử dụng các proxy miễn phí cho phép thay đổi bất kì dữ liệu nào được gửi từ trình duyệt. Hoặc tin tặc có thể tấn công trực tiếp bằng cách telnet gửi http header đến trình chủ.

```
su-2.05# telnet localhost 80
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0
Referer: <!--#virtual include="somefile.log"-->
User-Agent: <!--#exec cmd="/bin/id"-->
HTTP/1.1 200 OK
Date: Mon, 17 Dec 2001 20:39:02 GMT
Server:
Connection: close
Content-Type: text/html
```

Ví dụ 1: Referer header chứa URL của trang web mà từ đó yêu cầu được gửi đi. Vì thế một vài website sẽ kiểm tra header này để đảm bảo rằng nó được gửi từ trang web của website đó. Việc làm này dùng để ngăn chặn việc tin tặc lưu lại trang web xuống máy, chỉnh sửa thuộc tính form, phá hoại bằng cách nhằm vào client side validate hay server side include, sau đó gửi đi. Nhưng phương pháp kiểm tra này sẽ thất bại khi tin tặc có thể sửa lại Referer header để nó giống như được gửi từ trang web hợp lệ.

```
Referer: <!--#virtual include="somefile.log"-->
Referer: <thực thi lệnh java script để phá hoại trang web>
```

Ví dụ 2: Còn với Accept-Language header dùng để xác định ngôn ngữ người dùng sử dụng. Một ứng dụng mạng thực hiện việc quốc tế hoá ngôn ngữ bằng cách đặt label ngôn ngữ lên đầu HTTP header và chuyển nó tới database để có thể xem được dưới dạng text. Nếu nội dung của header được gửi nguyên mẫu tới database, khi đó tin tặc có thể dùng các câu lệnh SQL để sửa lại header. Nếu như thành phần header được dùng để xây dựng tên file để từ đó có thể xem đúng ngôn ngữ, lúc này tin tặc có thể sửa đổi để dẫn người sử dụng đến một nhánh khác.

```
Accept-Language "en, fr"
```

Ứng dụng web tìm tên ngôn ngữ trong HTTP header trong cơ sở dữ liệu. Chính vì vậy tin tặc có thể chèn lệnh SQL(SQL injection) vào bằng cách chỉnh sửa header. Tương tự nếu nội dung của header là một tên tập tin để từ đó sẽ tìm ra tên ngôn ngữ thì tin tặc có thể dùng cách tấn công path traversal.

Kĩ thuật phòng chống

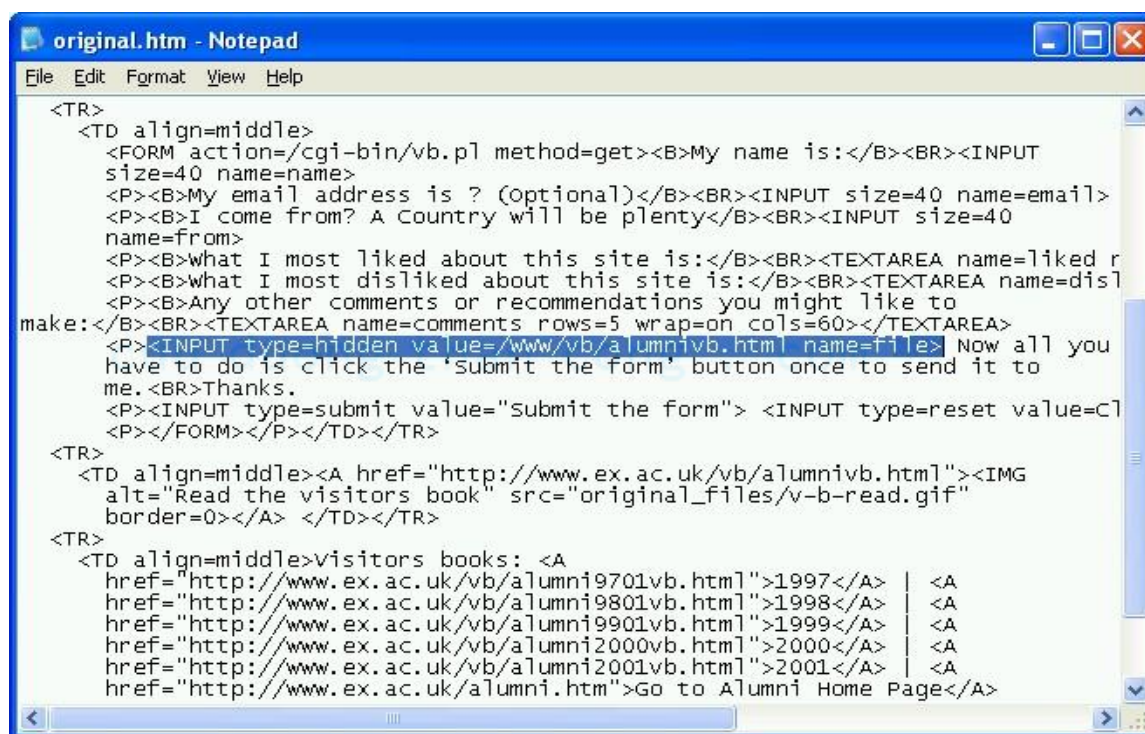
Đơn giản là không tin tưởng vào header nếu chưa có các biện pháp an toàn. Với các header gửi từ server chẳng hạn như cookie thì có thể được mã hoá. Còn với các header gửi từ client như referer thì không nên dùng chúng để thực hiện các biện pháp an toàn.

Không bao giờ để những giá trị nhạy cảm vào header

Không bao giờ cho người khác có thể xem các file thông qua các đoạn script.

Ví dụ :

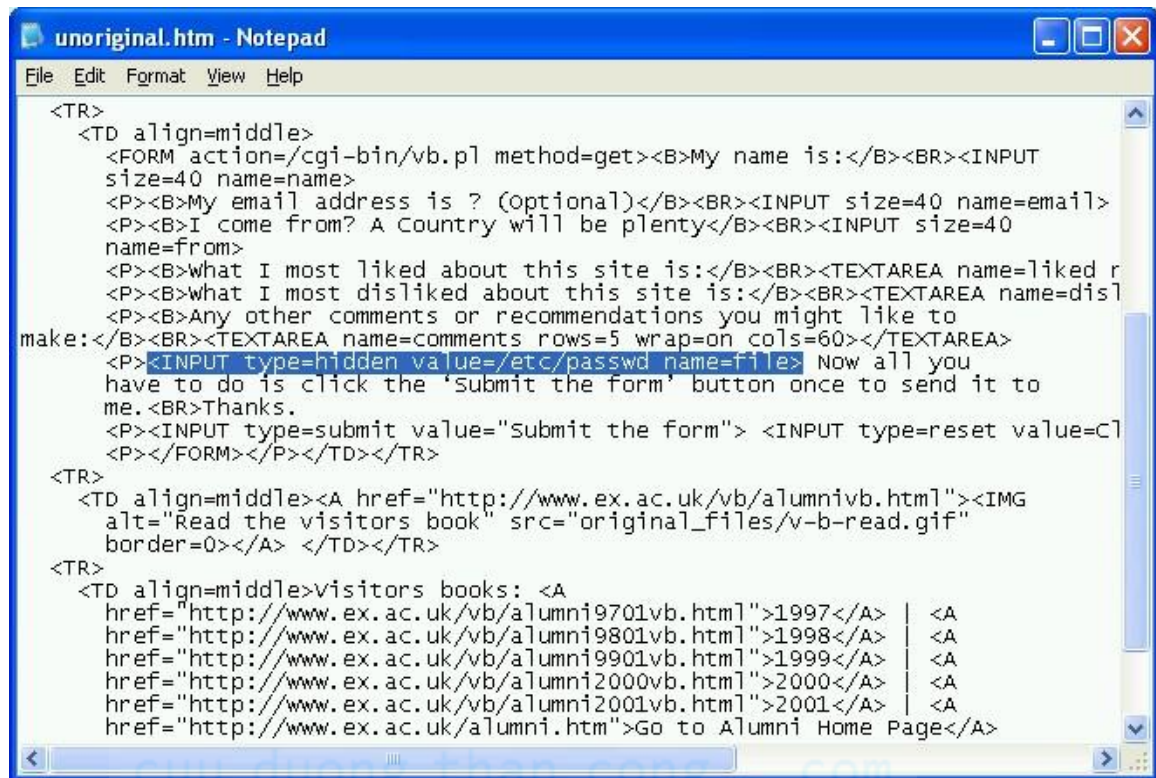
Header nguyên mẫu



```
original.htm - Notepad
File Edit Format View Help
<TR>
  <TD align=middle>
    <FORM action=/cgi-bin/vb.pl method=get><B>My name is:</B><BR><INPUT
    size=40 name=name>
    <P><B>My email address is ? (Optional)</B><BR><INPUT size=40 name=email>
    <P><B>I come from? A Country will be plenty</B><BR><INPUT size=40
    name=from>
    <P><B>what I most liked about this site is:</B><BR><TEXTAREA name=liked r
    <P><B>what I most disliked about this site is:</B><BR><TEXTAREA name=disl
    <P><B>Any other comments or recommendations you might like to
    make:</B><BR><TEXTAREA name=comments rows=5 wrap=on cols=60></TEXTAREA>
    <P><INPUT type=hidden value=/www/vb/alumnivb.html name=file> Now all you
    have to do is click the 'submit the form' button once to send it to
    me.<BR>Thanks.
    <P><INPUT type=submit value="submit the form"> <INPUT type=reset value=C1
    <P></FORM></P></TD></TR>
<TR>
  <TD align=middle><A href="http://www.ex.ac.uk/vb/alumnivb.html"><IMG
  alt="Read the visitors book" src="original_files/v-b-read.gif"
  border=0></A> </TD></TR>
<TR>
  <TD align=middle>visitors books: <A
  href="http://www.ex.ac.uk/vb/alumni9701vb.html">1997</A> | <A
  href="http://www.ex.ac.uk/vb/alumni9801vb.html">1998</A> | <A
  href="http://www.ex.ac.uk/vb/alumni9901vb.html">1999</A> | <A
  href="http://www.ex.ac.uk/vb/alumni2000vb.html">2000</A> | <A
  href="http://www.ex.ac.uk/vb/alumni2001vb.html">2001</A> | <A
  href="http://www.ex.ac.uk/alumni.htm">Go to Alumni Home Page</A>
```

cuuduongthancong.com

Header đã bị sửa đổi



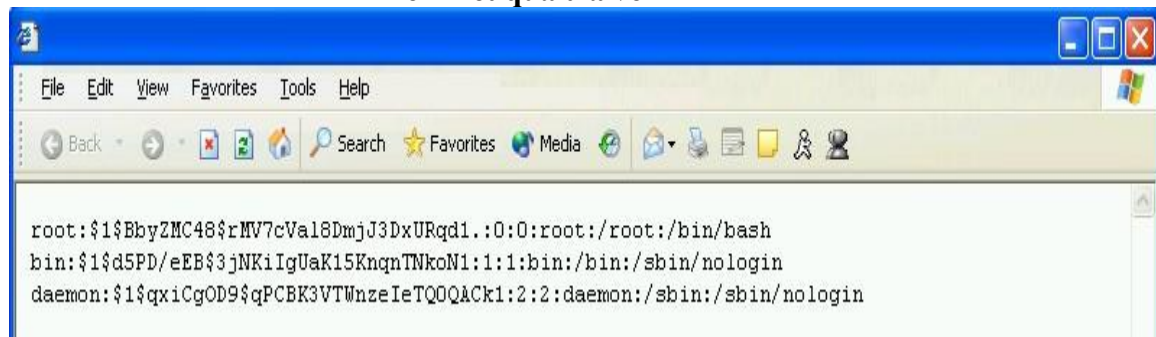
```
unoriginal.htm - Notepad
File Edit Format View Help

<TR>
  <TD align=middle>
    <FORM action=/cgi-bin/vb.pl method=get><B>My name is:</B><BR><INPUT
    size=40 name=name>
    <P><B>My email address is ? (Optional)</B><BR><INPUT size=40 name=email>
    <P><B>I come from? A Country will be plenty</B><BR><INPUT size=40
    name=from>
    <P><B>what I most liked about this site is:</B><BR><TEXTAREA name=liked r
    <P><B>what I most disliked about this site is:</B><BR><TEXTAREA name=disl
    <P><B>Any other comments or recommendations you might like to
    make:</B><BR><TEXTAREA name=comments rows=5 wrap=on cols=60></TEXTAREA>
    <P><INPUT type=hidden value=/etc/passwd name=file> Now all you
    have to do is click the 'submit the form' button once to send it to
    me.<BR>Thanks.
    <P><INPUT type=submit value="submit the form"> <INPUT type=reset value=C1
    <P></FORM></P></TD></TR>

<TR>
  <TD align=middle><A href="http://www.ex.ac.uk/vb/alumnivb.html"><IMG
  alt="Read the visitors book" src="original_files/v-b-read.gif"
  border=0></A> </TD></TR>

<TR>
  <TD align=middle>visitors books: <A
  href="http://www.ex.ac.uk/vb/alumni9701vb.html">1997</A> | <A
  href="http://www.ex.ac.uk/vb/alumni9801vb.html">1998</A> | <A
  href="http://www.ex.ac.uk/vb/alumni9901vb.html">1999</A> | <A
  href="http://www.ex.ac.uk/vb/alumni2000vb.html">2000</A> | <A
  href="http://www.ex.ac.uk/vb/alumni2001vb.html">2001</A> | <A
  href="http://www.ex.ac.uk/alumni.htm">Go to Alumni Home Page</A>
```

Xem kết quả trả về



```
root:$1$BbyZMC48$rMV7cVa18DmjJ3DxURqd1.:0:0:root:/root:/bin/bash
bin:$1$d5PD/eEB$3jNKiIgUaK15KnqnTNkoN1:1:1:bin:/bin:/sbin/nologin
daemon:$1$qxiCgOD9$qPCBK3VTWnzeIeTQOQ&Ck1:2:2:daemon:/sbin:/sbin/nologin
```

cuu duong than cong . com

4. Known Vulnerability Protection - Biết chỗ yếu bảo vệ

Đó là những thành phần dễ bị tấn công chẳng hạn như là các lỗi hay là các lỗ hổng bảo mật trong các hệ điều hành và các thành phần thứ ba. Do sử dụng những cấu hình có sẵn hay cấu hình sai hay không an toàn dễ dàng cho tin tặc tấn công.

Ta có thể sử dụng một số công cụ thường được dùng như:

- Nmap (www.insecure.org/nmap)
- Netcat
- Strobe (packetstorm.security.com)
- ISS (www.iss.net)
- Các trình quét cổng và các tài nguyên chia sẻ (share) khác dùng cho Windows như Superscan, Sechole, Redbutton, Net Essential...

Quá trình duyệt cổng có thể cho ta biết được các dịch vụ mạng nào được sử dụng. Ví dụ như các cổng TCP: 139, 135 (NETBIOS), 110 (pop3), 80 (HTTP), 79 (Finger), 53 (domain), 25 (smtp), 21 (ftp)...Thậm chí cả hệ điều hành và webserver.

Ta có thể tìm thông tin từ các website sau:

- www.securityfocus.com
- www.10pht.com
- www.microsoft.com/security packetstorm.security.com

Hoặc có thể đăng ký ở các mailing list để có thể nhận được các thông tin về security cập nhật nhất:

- Bugtraq (www.securityfocus.com)
- NTBugTraq (www.ntbugtraq.com)
- Pen-Test (www.securityfocus.com)

Một số chương trình dùng để phát hiện lỗi :

[1]. Grinder

Grinder 1.1 (ta có thể tải chương trình này từ

<http://tin.tacsclub.com/km/files/hfiles/rhino9/grinder11.zip>) của Rhino version 9 là chương trình ứng dụng Win32, chuyên quét dãy địa chỉ IP rồi báo cáo tên và số hiệu phiên bản của chính web server. Chẳng có gì khác so với lệnh HEAD (dùng netcat) nhưng Grinder tạo nhiều socket song song, vì vậy nó sẽ chạy rất nhanh.

Một cơ chế khác báo cáo phiên bản web server là kịch bản quét UNIX trên web site Hacking Exposed (www.osborne.com/hacking). Nếu đưa vào cổng 80 vào tập tin cổng, theo mặc định lệnh HEAD sẽ gửi đến web server rồi báo cáo tên và số hiệu phiên bản phần mềm đang chạy, để thông tin vào tập tin

<name>/<name>.http.dump. Muốn quét, ta có thể thực hiện bằng cú pháp sau:

```
/unixscan.pl hosts.txt ports.txt test -p -z  
-r -v
```

Sau khi hoàn tất, tập tin sẽ báo phiên bản Web Server

```
172.29.11.82 port 80: Server:  
Microsoft - IIS/4.0  
172.29.11.83 port 80: Server:  
Microsoft - IIS/3.0  
172.29.11.84 port 80: Server:  
Microsoft - IIS/4.0
```

[2]. Site Scan

SiteScan có mức quét sâu hơn Grinder một cấp, nó kiểm tra các chỗ yếu web cụ thể như PHF, PHP, finger, test.cgi,... Chương trình này chỉ lấy địa chỉ IP nên không thể đưa công cụ viết kịch bản. Ta cần phải tự nhập địa chỉ IP cụ thể rồi báo cáo kết quả.

5. Cross Site Scripting

Nhiều ứng dụng trên mạng sử dụng cookie để lưu thông tin trên máy khách (như user ID, thời gian kết nối ...). Do cookie không phải lúc nào cũng được mã hóa nên tin tặc có thể sửa đổi cookie để đánh lừa chương trình ứng dụng truy cập bất hợp pháp thông tin trong CSDL. Tin tặc cũng có thể ăn cắp cookie của một người dùng nào đó để truy cập thông tin của người này mà không cần phải biết user name và password.

6. Forceful Browsing

Các Forceful Browsing Web server sẽ gửi bất kì file nào cho user nếu như user biết tên file và file không được bảo vệ. Vì vậy tin tặc có thể khai thác lỗ hổng này để “nhảy” trực tiếp đến trang cần tấn công.

Cách thức phòng chống :

- Không bao giờ xây dựng HTML chung một nơi.
- Sử dụng một tiến trình ẩn khi tìm những file back up. Sử dụng một file đơn giản để chứa tất cả các ngoại lệ không cho truy cập tới, cách này rất hiệu quả.

- Một vài web server / application server xây dựng những trang động sẽ không trả về thông điệp 404 tới trình duyệt, nhưng nó sẽ trả về một trang định vị. Điều này làm cho các scanner lầm tưởng là tất cả các file vẫn tồn tại.
- Không trả lời các file được tham chiếu để bảo vệ trang web trước sự tấn công của những tin tặc.
- Xoá bỏ những file không cần thiết trên web server, chắc chắn rằng những file này không cần thiết.
- Có rất nhiều danh sách các file tồn tại, cho nên chúng ta có một số các link đưa đến trang trống ở đây.

7. Stealth Commanding - Chèn mã lệnh

Đây là kỹ thuật chèn mã lệnh vào trang Web từ máy khách. Kỹ thuật này cho phép tin tặc đưa mã lệnh thực thi vào phiên làm việc trên Web của một người dùng khác. Khi đoạn mã lệnh này chạy, nó sẽ cho phép tin tặc làm đủ thứ chuyện, từ giám sát phiên làm việc trên Web cho đến toàn quyền điều khiển máy tính của nạn nhân.

8. Backdoor and Debug Options

Khi viết chương trình các lập trình viên thường tạo các cửa hậu cũng như các tính năng bẫy lỗi để tiện cho việc kiểm tra và phát hiện lỗi. Việc này rất có ích trong quá trình phát triển sản phẩm ứng dụng. Tuy nhiên, các tính năng bẫy lỗi này lại thường không được bỏ đi khi hoàn tất sản phẩm vì một lý do nào đó và chúng trở thành lỗ hổng bảo mật dành cho tin tặc.

9. 3rd Party Misconfiguration - Cấu hình không an toàn

Đây là lỗ hổng do ứng dụng có các thiết lập mặc định không an toàn hoặc do người quản trị hệ thống định cấu hình không an toàn. Ví dụ như cấu hình Web server cho phép bất kỳ ai cũng có quyền duyệt qua hệ thống thư mục. Việc này có thể làm lộ các thông tin nhạy cảm như mã nguồn, mật khẩu hay thông tin của khách hàng.

10. Cookie Poisoning

Phương thức tấn công này tham chiếu đến cookie và sửa đổi dữ liệu của nó và quay trở lại dùng cookie bị sửa đổi để truy cập tới tài khoản đó.

11. SQL Injection

Kỹ thuật tấn công này lợi dụng những lỗ hổng trên ứng dụng (không kiểm tra kỹ những kí tự nhập từ người dùng). Thực hiện bằng cách thêm các mã vào các câu lệnh hay câu truy vấn SQL (thông qua những textbox) trước khi chuyển cho ứng dụng web xử lý, Server sẽ thực hiện và trả về cho trình duyệt (kết quả câu truy vấn hay những thông báo lỗi) nhờ đó mà các tin tặc có thể đăng nhập mà không cần username và password, điều hành từ xa, kết xuất, xoá, sửa cơ sở dữ liệu và lấy root của SQL server.

Tài liệu tham khảo :

- [1]. **Building Internet Firewall** – O Reilly – D.Brent Chapman & Elizabeth D. Zwicky
- [2]. **AFITC 2001 Web Application Security** by Jeremiah Grossman.
- [3]. **Hacking Exposed Second Edition** - Joel Scambray, Stuart McCure, George Kurtz

cuu duong than cong . com

cuu duong than cong . com